NASA CR 114381
AVAILABLE TO THE PUBLIC

METHODOLOGIES FOR OPTIMAL RESOURCE ALLOCATION TO THE NATIONAL SPACE PROGRAM AND NEW SPACE UTILIZATIONS

FINAL REPORT

VOLUME 2

PROGRAMMER'S MANUAL
RESOURCE ALLOCATION AND SMOOTHING MODEL

19 NOVEMBER 1971

PREPARED UNDER CONTRACT NAS2-5202

FOR

ADVANCED CONCEPTS AND MISSIONS DIVISION
OFFICE OF ADVANCED RESEARCH AND TECHNOLOGY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

BY

LOCKHEED MISSILES & SPACE COMPANY
SUNNYVALE, CALIFORNIA

"Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the authors or organization that prepared it."

NASA CR 114381

METHODOLOGIES FOR OPTIMAL RESOURCE ALLOCATION TO THE NATIONAL SPACE PROGRAM AND NEW SPACE UTILIZATIONS

FINAL REPORT

VOLUME 2

PROGRAMMER'S MANUAL
RESOURCE ALLOCATION AND SMOOTHING MODEL

19 NOVEMBER 1971

Prepared Under Contract NAS2-5202

For

ADVANCED CONCEPTS AND MISSIONS DIVISION
OFFICE OF ADVANCED RESEARCH AND TECHNOLOGY
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA

By
LOCKHEED MISSILES & SPACE COMPANY
SUNNYVALE, CALIFORNIA

FOREWORD

This report volume is the programmer's manual developed during a study of methodologies and systems modeling for optimal assignment of resources to the national space program and for the evaluation of potential new space directions. This study is being performed for the National Aeronautics and Space Administration under Contract NAS2-5202, and is monitored by Mr. R. E. Slye and Mr. Harold Hornby of the Advanced Concepts and Missions Division of the Office of Advanced Research and Technology.

Individuals of Lockheed Missiles & Space Company who contributed to this study are L. F. Fox, project leader; C. J. Golden, key technical member; and W. T. Lew

CONTENTS

SUMMARY

This document is Volume 2 of a two volume series entitled Methodologies for Optimal Resource Allocation to the National Space Program and New Space Utilizations. This volume is a programmer's manual for the optimal resource allocation and budget smoothing model described in Volume 1.

This volume contains appendixes that provide model input requirements, a sample case, flow charts, and a program listing. At the beginning of each appendix, descriptive details and technical comments are provided to indicate any special instructions applicable to the use of that appendix. In addition, the program listing of Appendix D includes comment cards that state the purpose of each subroutine in the complete program and describe operations performed within that subroutine.

Appendix A, Input Requirements, provides details on the many options that adapt the program to the specific needs of the analyst for a particular problem.

Appendix A

INPUT REQUIREMENTS

A.1 GENERAL

A complete glossary of input terms and detailed format requirements are included in this appendix. Variable names are listed by order of input in corresponding sections of use to make the glossary easier to use than an alphabetical listing. Comments are also included which describe either external or internal restrictions associated with the variable.

Figure A-1 illustrates the basic data deck layout for this program. Any section may be eliminated if there are no associated data. However, either a blank card must be inserted in place of the section or the control card must reflect no input for that section. If the control card is coded so no data are input for some section, then values input for the preceding case are automatically supplied. Otherwise, if no data are desired for any one section then a blank card must replace that section followed by a blank card which designates the end of that section. Stage performance data to be used in the stage-matching screen may be eliminated entirely, including the final blank card, if the stage-matching screen is not to be used. If this screen is used, then the stage cards must be ordered so that all stages in Class 1 precede those in Class 2, which precede those in Class 3, which precede those in Class 4. Stages not included in the matching screen follow those in the above classifications. If the matching screen is not used, the order of cards within each section is unimportant.

Constraint and budget level cards are input to the SMOOTH subroutine of integrated program. The last data card input to this budget section is followed by a card containing only an asterisk in the first column.

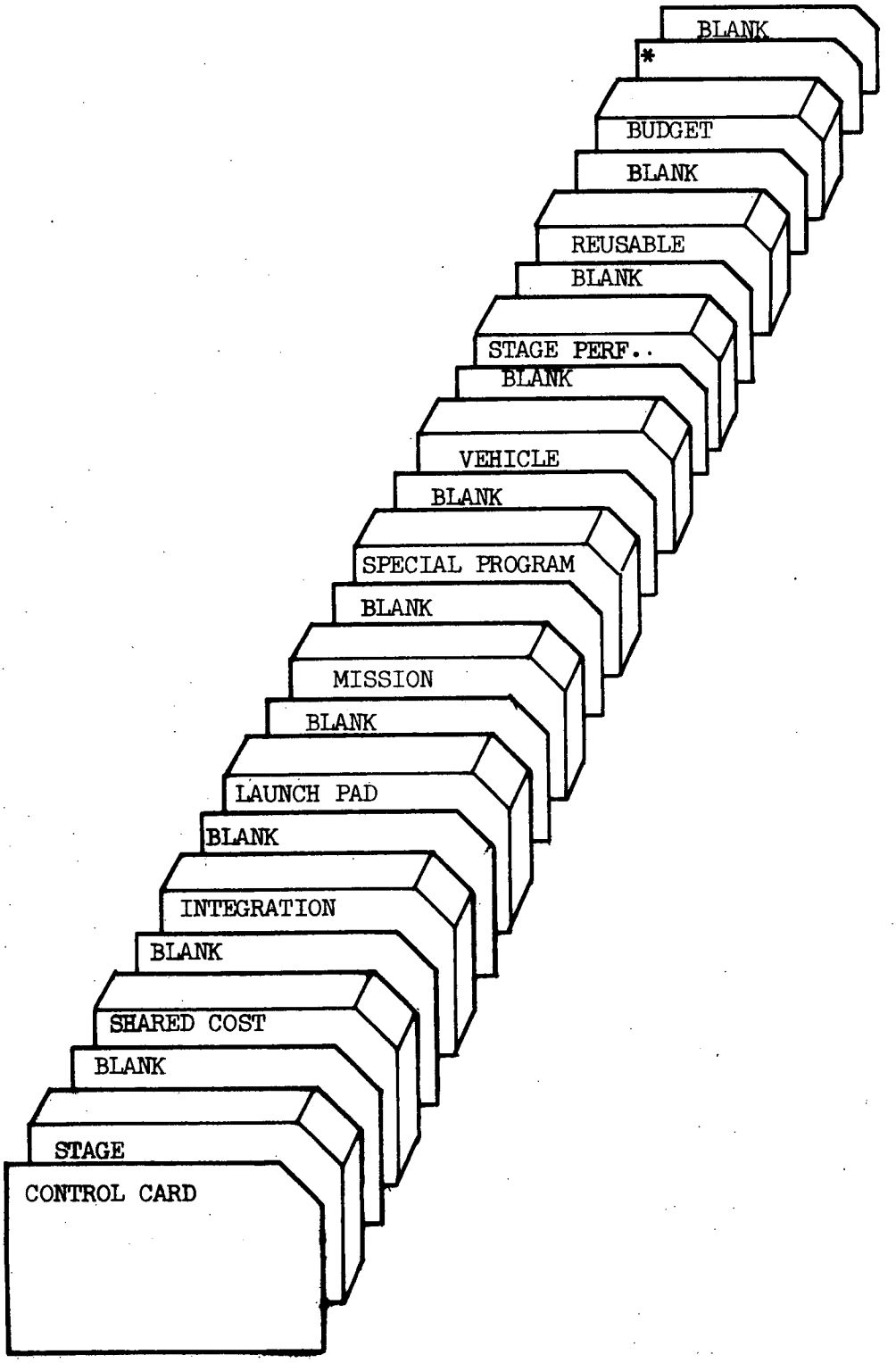


Figure A-1 Data Deck Layout

Then the control card for the next set of data appears unless there are no more data cases to follow. In this latter case, a blank card follows the asterisk card in order to terminate the run under normal circumstances.

A.2 INPUT FORM AND DEFINITIONS

Card Column	Variable Name	Format	Description and Comments
Control Card			
1 - 3	LP	I3	Code for logic printout If LP $\geq$ 2 Print decision numbers for each vehicle If LP $\geq$ 1 Print logic associated with algorithm If LP = 0 No logic output
4 - 6	NOPT	I3	Code for mission/vehicle compatibility screen 1 - ΔV vs. payload weight + availability + a priori assignment 2 - Code 1 plus use stage-matching screen 3 - All criteria
7 - 9	MOS	I3	Method of solution desired MOS = 0 Optimize assignment and smooth resulting budget MOS = 1 Input assignment and smooth resulting budget MOS = 2 Optimize assignment and output associated costs MOS = 3 Input assignment and print out associated costs
10 - 12	NSOL	I3	Number of solutions to be output in ascending order of total program cost

Card Column	Variable Name	Format	Description and Comments
13 - 15	MSOL	I3	MSOL = 1 bypasses time-saving feature in algorithm
16 - 18	MITR	I3	Maximum number of allowed iterations between SMOOTH AND ASSIGN
19 - 21	ILY	I3	Last two digits of initial calendar launch year of mission model
22 - 24	MYRS	I3	Mission model duration in years
25 - 29	TREF	F5.1	Last 2 digits of calendar year for SMOOTH
30 - 41	GUESS	F12.2	Upper bound for total launch vehicle program (saves storage space if realistic value). If GUESS = 0.0, then GUESS is assigned a value 1.0 E15
42 - 44	GRO	F3.1	Annual economic growth factor, e.g., 7% inflation/year; GRO = 7 .
45 - 49	SLO	F5.1	Annual Sustaining Costs less than SLO are left out of the basic algorithm and treated later in determining the optimal assignment. MSOL = 0 option must be specified for implementation
50 - 53	CORR	F4.2	Correlation between development and operating cost growth factors (e.g., 0.3)
67 - 68	IP	12	Code for pad input*
69 - 70	IG	12	Code for stage input*
71 - 72	IFM	12	Code for shared cost group input*
73 - 74	II	12	Code for integration cost input*
75 - 76	IM	12	Code for mission input*
77 - 78	ISD	12	Code for special program data*
79 - 80	IV	12	Code for vehicle input*

* If $\geq$ 0, new input for this case

If < 0, use data from previous case

Card Columns	Variable Name	Format	Description and Comments
Stage Information (Input only if IG $\geq$ 0) I = 1, NSTG < 50			
1 - 2	KODS(I)	I2	Reference number of stage on card I
4 - 7	STG(I)	A4	Name of stage on card I
8 - 13 14 - 19 20 - 25	SR(I,J) J = 1,3	3F6.3	Recurring cost for first unit of stage on card I J = 1 Hardware J = 2 ETR launch support J = 3 WTR launch support
26 - 30 31 - 35 36 - 40	PLC(I,J) J = 1,3	3F5.3	Recurring cost learning curve percent for stage on card I in decimal form (e.g., .95)† J = 1 Hardware J = 2 ETR launch support J = 3 WTR luanch support
44 - 49	SNR(I)	F6.3	Development cost of stage on card I
50 - 55	STS(I)	F6.3	Sustaining cost of stage on card I
59 - 61	LSA(I)	I3	Last year stage on card I is available*+
62 - 64	NBY(I)	I3	Batching duration in years for stage recurring cost
65 - 67 68 - 70 71 - 73 74 - 76	NFS(I,J) J = 1,4	4I3	KODEF of the shared cost groups (up to 4) to which stage belongs

† If percent is 100, then input zero for more efficient program operation

* 1 corresponds to year ILY

\+ If available through mission model, any number $\geq$ MYRS may be input. If number < MYRS is input then this termination date is maintained through all iterations.

Card Column	Variable Name	Format	Description and Comments
78 79 80	MODE(I,J) J = 1,3	3I1	Code to indicate type of input for recurring cost of stage on card I* J = 1 Hardware J = 2 ETR launch support J = 3 WTR launch support

Second Stage Card

5 - 9 10 - 14	SUSLS(I,J) J = 1,2	2F5.0	Sustaining cost at launch facility for Stage I, not to be duplicated at each pad. J = 1 ETR J = 2 WTR
15 - 17	NU(I)	I3	Number of reusable units in initial investment of component I NU = 0 unit is expendable NU > 0 estimate used by program directly NU $\leq$ -2 estimate used by program for first iteration, then subroutine REUSE calculates estimate for NU
18 - 23	UPP(I)	F6.2	Unit purchase price
24 - 29 30 - 32	UPPXX PXX	F6.1 F3.2	PXX% tail such that using the lognormal distribution, prob.(UPP(I) $\geq$ UPPXX) = PXX (e.g., PXX = .05)
33 - 38	RPLO(I)	F6.0	Return payload weight in lbs for this component.** (Vehicle return payload = orbiter return payload)
39 - 40	YDS(I)	F2.0	Duration in years over which β function distributes development cost for stage on card I. (Leave blank if SNR(I) = 0 . Input necessary if SNR $\neq$ 0)

* If = 0, learning curve type input
If = 0, jump type input

** Any value $\geq$ 1.0 may be input to indicate payload return capability besides crew.

Card Column	Variable Name	Format	Description and Comments
41 - 42	IST(I)	I2	Last 2 digits of calendar start date for Stage Development Program (necessary if SNR or STS≠0)
43 - 44	NSFX(NSDC*)	I2	Duration in years ≤ 12 for any miscellaneous (fixed or development) program associated with stage on card I (e.g., Run out costs). (Standard Development costs are distributed by a Beta function - any other development distribution may be input under this special category.)
45 - 49 50 - 54 55 - 59	SRXX(J) J = 1,3	3F5.1	XX% tail such that prob.(SR(J) ≥ SRXX(J)) = XX(1) for J = 1,3
60 - 62	XX(1)	F3.2	Percent tail above in decimal form (e.g., .05)
63 - 68 69 - 71	SNRXX XX(2)	F6.1 F3.2	XX% tail such that prob. (SNR(I) ≥ SNRXX) = XX(2)
72 - 77 78 - 80	STSXX XX(3)	F6.1 F3.2	XX% tail such that prob. (STS(I) ≥ STSXX) = XX(3)

If MODE(I,J) ≠ 0 for some J, require following Jump Type Input Card for Each such J.

5 - 14	SRJ(LX,1)	F10.3	Total recurring cost for up to POJ number of stages
15 - 24	SRJ(LX,2)	F10.3	Slope of line defining total recurring cost for over POJ number of stages
25 - 34	SRJ(LX,3)	F10.3	Y-intercept of line defining total recurring cost for over POJ number of stages
35 - 44	POJ(LX)	F10.3	Number of stages at which function defining total recurring cost changes slope

* NSDC = Number of special development costs ≤ 50

Card Column	Variable Name	Format	Description and Commments
45 - 54	SRJXX	F10.3	PXX% tail such that
55 - 57	PXX	F3.2	prob. (SRJ(LX,1) $\geq$ SRJXX) = PXX

If NSFX(NSDC) $\neq$ 0 read in following card.

1 - 3	NRFX(NSDC)	I3	Start date for special development cost associated with stage on card I. (Referenced to IST(I))
4 - 8 9 - 13 14 - 18 19 - 23 24 - 28 29 - 33 34 - 38 39 - 43 44 - 48 49 - 53 54 - 58 59 - 63	RXD(J,NSDC) J = 1,12	12F5.2	Special development cost to be spent in calendar year 1900 + IST(I) + NRFX(NSDC) - 2 + J (Input distribution)
64 - 69	RXDXX	F6.1	PXX% tail such that
70 - 72	PXX	F3.2	prob($\sum_{j=1}^{12}$ RXD(J,NSDC) $\geq$ RXDXX) = PXX

Last Stage Card must be followed by a blank card.

Shared Cost Group Cards (Input only if IFM $\geq$ 0) I = 1, NFAM < 40

1 - 2	KODEF(J)=I	I2	Reference Number of group on card J
4 - 7	FAM(I)	A4	Name of group I
8 - 17	FMNR(I)	F10.0	Development cost of group I
18 - 27	FMSUS(I)	F10.0	Sustaining cost of group I

Card Column	Variable Name	Format	Description and Comments
28 - 31	YDF(I)	F4.1	Duration in years of Development Program cost distribution (β Function). (Leave blank if FMNR(I) = 0.)
32 - 34	JST(I)	I3	Last 2 digits of calendar start date for group Development Program - necessary if FMNR or FMSUS $\neq$ 0
35 - 37	NSFX(NSDC)	I3	Duration in years for any miscellaneous fixed or development program distribution associated with group I. (Distribution input on following card.)
38 - 47 48 - 57	FMSLS(I,J) J = 1,2	2F10.0	Sustaining cost at launch site for group I not to be duplicated at each pad. J = 1 ETR J = 2 WTR
58 - 64 65 - 67	FMNRXX XX(1)	F7.0 F3.2	XX% tail such that prob.(FMNR(I) $\geq$ FMNRXX) = XX(1)
68 - 74 75 - 77	FMSSXX XX(2)	F7.0 F3.2	XX% tail such that prob.(FMSUS(I) $\geq$ FMSSXX) = XX(2)

If (NSFX(NSDC) $\neq$ 0) read following card.

1 - 3	NRFX(NSDC)	I3	Start date for special Development cost associated with group I. (Referenced to JST(I)).
4 - 8 9 - 13 14 - 18 ⋮ 59 - 63	RXD(J,NSDC) J = 1,12	12F5.2	Special Development cost to be spent in calendar year 1900 + JST(I) + NRFX(NSDC) - 2 + J (Input distribution)
64 - 69 70 - 72	RXDXX PXX	F6.1 F3.2	PXX% tail such that prob.$\left(\sum_{j=1}^{12} \text{RXD}(J,\text{NSDC}) \geq \text{RXDXX}\right) = \text{PXX}$

Card Column	Variable Name	Format	Description and Comments
Last Group card must be followed by a blank card.			
Integration Cost Cards (Input only if II ≥ 0) I = 1, NCI < 30			
3 - 5	NFML(I)	I3	KODEF of shared cost group which is lower member of integration pair I
6 - 8	NFMU(I)	I3	KODEF of shared cost group which is upper member of integration pair I
9 - 18	RINT(I)	F10.0	Recurring cost for first unit of integration I
19 - 28	PLCINT(I)	F10.0	Recurring cost learning curve percent for integration I
29 - 38	DINT(I)	F10.0	Development cost of integration I
39 - 48	SINT(I)	F10.0	Sustaining cost of integration I
49 - 52	YDI(I)	F4.1	Development duration in years for β distribution (Leave blank if DINT(I) = 0.)
53 - 55	KST(I)	I3	Last 2 digits of calendar start date for integration development program - input necessary if DINT or SINT ≠ 0
56 - 58	NSFX(NSDC)	I3	Duration in years for any miscellaneous fixed or development program associated with integration I. (Distribution input on following card).
59 - 68	SINTLS(I,J)	2F10.0	Sustaining cost at launch facility for integration I not to be duplicated at each pad.
69 - 78	J = 1,2		J = 1 ETR J = 2 WTR
Second Integration Card			
1 - 10	RINTXX	F10.0	XX(1)% tail such that prob.(RINT(I) ≥ RINTXX) = XX(1)
11 - 13	XX(1)	F3.2	

Card Column	Variable Name	Format	Description and Comments
14 - 23	DINTXX	F10.0	XX(2)% tail such that prob.(DINT(I) $\geq$ DINTXX) = XX(2)
24 - 26	XX(2)	F3.2	
27 - 36	SINTXX	F10.0	XX(3)% tail such that prob.(SINT(I) $\geq$ SINTXX) = XX(3)
37 - 39	XX(3)	F3.2	

IF NSFX(NSDC) $\neq$ 0 read following card.

1 - 3	NRFX(NSDC)	I3	Start date for Special Development cost associated with integration I. (Referenced to KST(I))
4 - 8 9 - 13 14 - 18 ⋮ 59 - 63	RXD(J,NSDC) J = 1,12	12F5.2	Special Development cost to be spent in calendar year 1900 + KST(I) + NRFX(NSDC) - 2 + J (Input distribution)
64 - 69	RXDXX	F6.1	PXX% tail such that
70 - 72	PXX	F3.2	$\text{prob}\left(\sum_{j=1}^{12} \text{RXD}(J,\text{NSDC}) \geq \text{RXDXX}\right) = \text{PXX}$

Last Integration card must be followed by a blank card.

Pad Cards (Input only if IP $\geq$ 0) I = 1, NP $\leq$ 30

1 - 4	KODEP(I)	I4	Number of pad complex on card I
7 - 10	PAD(I)	A4	Name of complex on card I
11 - 15	NPERPD(I)	F5.0	Maximum number of launches/year/pad possible at complex I

2nd - 6th cards needed for each pad complex (stage cost data, 2/card) (J = 1,10)

3 - 4 43 - 44	NPSTG(I,J)	I2	KODE corresponding to Jth stage costs of complex I

Card Column	Variable Name	Format	Description and Comments
5 - 9 45 - 49	PSTGD(I,J,1)	F5.0	J^{th} stage development cost of first pad in complex I
10 - 12 50 - 52	YDPS(I,J)	F3.0	Development duration in years for β distribution
13 - 15 53 - 55	MST(I,J)	I3	Last 2 digits of calendar start date of PSTGD
16 - 20 56 - 60	PSTGS(I,J,1)	F5.0	J^{th} stage sustaining cost of first pad in complex I
21 - 25 61 - 65	PSTGD(I,J,2)	F5.0	J^{th} stage development cost of second pad in complex I
26 - 30 66 - 70	PSTGS(I,J,2)	F5.0	J^{th} stage sustaining cost of second pad in complex I
31 - 35 71 - 75	PSTGD(I,J,3)	F5.0	J^{th} stage development cost of third pad in complex I
36 - 40 76 - 80	PSTGS(I,J,3)	F5.0	J^{th} stage sustaining cost of third pad in complex I

7th - 9th cards needed for each pad complex (family cost data, 2/card)(J = 1,5)

3 - 4 43 - 44	NPFAM(I,J)	I2	KODEF corresponding to J^{th} family costs of complex I
5 - 9 45 - 49	PFAMD(I,J,1)	F5.0	J^{th} family development cost of first pad in complex I
10 - 12 50 - 52	YDPF(I,J)	F3.0	Development duration in years for β distribution
13 - 15 53 - 55	LST(I,J)	I3	Last 2 digits of calendar start date of PFAMD
16 - 20 56 - 60	PFAMS(I,J,1)	F5.0	J^{th} family sustaining cost of first pad in complex I

Card Column	Variable Name	Format	Description and Comments
21 - 25 61 - 65	PFAMD(I,J,2)	F5.0	J^{th} family development cost of second pad in complex I
26 - 30 66 - 70	PFAMS(I,J,2)	F5.0	J^{th} family sustaining cost of second pad in complex I
31 - 35 71 - 75	PFAMD(I,J,3)	F5.0	J^{th} family development cost of third pad in complex I
36 - 40 76 - 80	PFAMS(I,J,3)	F5.0	J^{th} family sustaining cost of third pad in complex I

10th and 11th cards needed for each pad complex (integration cost data, 3/card) (J = 1,5)

9 - 11 33 - 35 57 - 59	NPINTL(I,J)	I3	KODEF of lower group corresponding to J^{th} integration cost of complex I
12 - 14 36 - 38 60 - 62	NPINTU(I,J)	I3	KODEF of upper group corresponding to J^{th} integration cost of complex I
15 - 32 39 - 56 63 - 80	PINTS(I,J,K) K = 1,3	3F6.0	J^{th} integration sustaining cost of K^{th} pad in complex I

Last Pad Card must be followed by Blank Card.

Mission Data Card - (Input only if IM $\geq$ 0) I = 1, NMIS < 50

1 - 2	KODEM(I)	I2	Reference number of mission on card I
3 - 8	NAME(I)	A6	Name of mission on card I
9 - 12	PB(I)	F4.2	Priority of mission on card I
15 - 16	NSYR(I)	I2	Number of sustaining years required for SUS(I) after last launch year

Card Column	Variable Name	Format	Description and Comments
17 - 18	NYRSFX(I)	I2	Duration in years of any fixed or special development cost distribution associated with mission KODEM(I)
19 - 25	VLR(I)	F7.0	Characteristic velocity required in fps to accomplish mission on card I
26 - 28	RPLM(I)	F3.0	Return payload weight in lbs required by mission on card I*
29 - 31	TAMT(I)	F3.0	Number of days orbiter required for mission completion (only required if NU < 0 for some reusable stage).
32 - 38	WPR(I)	F7.0	Payload weight in lbs required for mission on card I
39 - 40	NTRIP(I)	I2	Maximum number of launches allowed to carry WPR(I) lbs into orbit. NTRIP(I) = 0 is same as 1.
41 - 80	MISN(I,J) J = 1,MYRS	20I2	Number of launches for mission on card I in calendar year J + 1900 + ILY - 1 with WPR payload at each launch

Second Mission Card

3 - 12	PLR(I)	F10.2	Payload recurring cost for mission KODEM(I).
13 - 22	SUS(I)	F10.2	Payload sustaining cost
23 - 32	C(I)	F10.2	Payload development cost
33 - 37	YDPL(I)	I5	Duration in years over which development cost is to be distributed by Beta Function
38 - 42 43 - 47 48 - 52 53 - 57	RDIST(I,L) L = 1,4	4F5.3	Input recurring cost distribution for PLR in decimal form (e.g., RDIST(I,1) = .05)

* Currently, any value $\geq$ 1.0 indicates that mission I is to be performed only by vehicles whose upper stage has RPLO > 1.0 .

Card Column	Variable Name	Format	Description and Comments
58 - 67	PLMD(I)	F10.2	Maximum diameter of payload for mission on card I
68 - 69	NPLS(I)	I2	Code for payload stabilization requirement 0 - No requirement 1 - Must be spin stabilized 2 - Must not be spin stabilized
70 - 71	MR(I)	I2	Code for man-rating requirement for mission on card I 0 - No requirement 1 - Must be man-rated
72 - 73	LTR(I)	I2	Code for launch site of mission 1 - ETR 2 - WTR
74 - 75	NRR(I)	I2	Number of restarts required for mission
76 - 77	IS(I)	I2	Last 2 digits of calendar start year for development cost PLD(I)
78 - 80	IVEHIA(I)	I3	A priori vehicle assignment for mission on card I If no vehicle assigned - 0 input; KODEV of vehicle input otherwise

Third Mission Card

1 - 10	PLRXX	F10.0	XX(1)% tail such that prob(PLR(I) $\geq$ PLRXX) = XX(1)
11 - 13	XX(1)	F3.2	
14 - 23	CXX	F10.0	Same for C(I)
24 - 26	XX(2)	F3.2	
27 - 36	SUSXX	F10.0	Same for SUS(I)
37 - 39	XX(3)	F3.2	

Card Column	Variable Name	Format	Description and Comments
If NYRSFX(I) $\neq$ 0 read following card.			
1 - 3	NSTRFX(I)	I3	Start date for special development cost associated with mission KODEM(I) referenced to IS(I)
4 - 8 9 - 13 14 - 18 ⋮ 59 - 63	RFIXD(J,I) J = 1, 12	12F5.2	Special Development cost to be spent in calendar year 1900 + IS(I) + NSTRFX(I) - 2 + J (Input distribution)
64 - 69	RXDXX	F6.1	PXX% tail such that
70 - 72	PXX	F3.2	$\text{Prob}\left(\sum_{j=1}^{12} \text{RFIXD}(J,I) \geq \text{RXDXX}\right) = \text{PXX}$

Last mission card must be followed by a blank card.

Special Program Data Card (No launch associated with program) - Input only if ISD $\geq$ 0, I = 1, NSPR $\leq$ 6

1 - 3	KODESP(I)	I3	Code number for Special Program (must be larger than 100)
4 - 9	NAME(I)	A6	Name of Special Program on card I
10 - 19	C(I)	F10.2	Development cost associated with program (distributed by β Function)
20 - 24	YDPL(I)	I5	Duration in years of Development program
25 - 26	IS(I)	I2	Last 2 digits of start year for development cost C(I)
27 - 36	SUS(I)	F10.2	Annual sustaining cost associated with program
37 - 38	NYRSST(I)	I2	Duration in years of sustaining program

Card Column	Variable Name	Format	Description and Comments
39 - 40	NYRSFX(I)	I2	Duration in years of any fixed cost which does not have a β distribution
41 - 50	CXX	F10.2	XX(1)% tail such that Prob(C(I) ≥ CXX) = XX(1)
51 - 53	XX(1)	F3.2	
54 - 63	SUSXX	F10.2	XX(2)% tail such that Prob(SUS(I) ≥ SUSXX) = XX(2)
64 - 66	XX(2)	F3.2	

If NYRSFX(I) ≠ 0 read following card.

1 - 3	NSTRFX(I)	I3	Start date for fixed cost referenced to IS(I)
4 - 8 9 - 13 14 - 18 ⋮ 59 - 63	RFIXD(J,I) J = 1,12	12F5.2	Fixed Cost to be spent in calendar year 1900 + IS(I) + NSTRFX(I) - 2 + J
64 - 69	RXDXX	F6.1	PXX% tail such that
70 - 72	PXX	F3.2	$\text{Prob}\left(\sum_{j=1}^{12} \text{RFIXD}(J,I) \geq \text{RXDXX}\right) = \text{PXX}$

Last Special Program Data card must be followed by a blank card.

Vehicle Data Card (Input only if IV ≥ 0) J = 1, NV ≤ 60

1 - 8	VEH(I,J) I = 1,4	4I2	KODS of stage in I^{th} position, where I = 1 corresponds to booster, for vehicle on card J
9 - 21	B1(J)	E13.6	Payload vs. characteristic velocity curve constants for performance evaluation of vehicle on card J. PL = EXP(B1 - B2*V - B3/(B4 - V)) and V = Excess Velocity = Total Characteristic Velocity-Circular Velocity at 100 n.m.
22 - 34	B2(J)	E13.6	
35 - 47	B3(J)	E13.6	
48 - 60	B4(J)	E13.6	

Card Column	Variable Name	Format	Description and Comments
79 - 80	KODEV(J)	I2	Reference number of vehicle on card J

2nd Card needed for each vehicle.

4 - 5	NVS(J)	I2	Code for stabilization of vehicle on card J 1 - Is spin stabilized 2 - Is not spin stabilized
6 - 7	MRV(J)	I2	Code for man-rating of vehicle on card J 0 - Is not man-rated 1 - Is man-rated
8 - 9	NRP(J)	I2	Number of restarts possible for vehicle on card J
10 - 12	NPAD(1,J)	I3	KODEP of pad complex at ETR from which vehicle J can be launched
13 - 15	NPAD(2,J)	I3	KODEP of pad complex at WTR from which vehicle J can be launched
16 - 18	NYP(1,J)	I3	1st year J^{th} vehicle can be flown from ETR
19 - 21	NYP(2,J)	I3	1st year J^{th} vehicle can be flown from WTR
80	JKEY	I1	Code for recurring cost distribution for vehicle on card J JKEY = 0 - standard distribution is used 1st year of distribution = .05 Recurring cost 2nd year of distribution = .20 Recurring cost 3rd year of distribution = .50 Recurring cost

(Continued on following page)

Card Column	Variable Name	Format	Description and Comments
			4th year of distribution = .25 Recurring cost = Launch year generating this recurring cost JKEY = 1 - Distribution is to be input on following card

Optional 3rd card for each vehicle (Input only if JKEY ≠ 0).

4 - 8 9 - 13 14 - 18 19 - 23	ALPI(I,J) I = 1,4	4F5.2	Input Recurring cost distribution for vehicle on card J in year I where I = 4 corresponds to year of launch

Last vehicle card must be followed by blank card.

Stage Performance Cards (Input only if NOPT = 2 on control card) I ≤ NSTG ≤ 40

1 - 4	KODE(I)	I4	Reference number of stage on card I (used to check order of cards)
5 - 9	NST(I)	I5	Classification of stage on card I
10 - 19	THRT(I)	F10.0	Stage thrust
20 - 29	DIAM(I)	F10.0	Stage diameter
30 - 39	TSL(I)	F10.0	Stage sea-level thrust
40 - 49	LENT(I)	F10.0	Interstage length required to clear engines
50 - 59	WTFU(I)	F10.0	Stage fuel weight
60 - 69	WTIN(I)	F10.0	Stage total inert weight
70 - 79	ISP(I)	F10.0	Stage vacuum specific impulse

Last Performance card must be followed by a blank card. Eliminate blank card if NOPT ≠ 2.

Reusable Stage Cards (One card required for each stage I with input negative NU(I)). (No special order required)

1 - 2	KODE(I)	I2	Stage code number of corresponding reusable stage

Card Column	Variable Name	Format	Description and Comments
5 - 6	NOB(I)	I2	Code to identify type of stage NOB = 1: BOOSTER NOB = 2: ORBITER
7 - 12	XLT(I)	F6.1	Amortization Lifetime (number of launches per unit before replacement)
13 - 18	TAT(I)	F6.1	Land to launch turn-around-time for first refurbishment (days)
19 - 24	PLCT(I)	F6.3	Learning curve percent for TAT in decimal form calculated from reference year (if zero is input, then PLCT is assumed to be 100)

Last Reusable Stage card must be followed by blank card. (Blank card required even if no data are input in this section.)

Budget Smoothing Data is input in subroutine SMOTHS using a CALL INPUT statement. The following variables may be input at this time.

Variable Name	Description and Comments
TITLE(I)	Output page HEADING - if no input blanks are output. 40 characters are allocated for storage, e.g., TITLE = 'LUNAR OPTION'
LEVEL(J)	YEARLY DESIRED FUNDING LEVEL (20 year maximum), e.g., LEVEL = 300., 375., 18 x 300. (in dollars of applicable year)
ISTRT	FIRST YEAR of smoothing interval - referenced to TREF = 1
IFIN	Last year of smoothing interval - referenced to TREF
MAXITR	Maximum number of iterations allowed per case in SMOTHS subroutine
NCSTR	Number of constraints on mission programs $\leq$ 90
NPROG(K)	The reference number (KODEM or KODESP) of the mission being constrained

Variable Name	Description and Comments
KPROG(K)	The reference number (KODEM or KODESP) of the constraining program or mission
KODE(K)	Code number for type of constraint $\leq$ 11
CS(K)	Constant associated with each constraint
FIXED(I)	Yearly total fixed overhead costs (I = 1,20) (in reference year dollars) If no input, is set to zero
PMAX, PMIN	Constants associated with PLOT2 - if no input they are set to 5000. and 1500. respectively
ACCL	Code for use of acceleration option - if no input it is set = TRUE
EXT	Code for use of extension option - if no input it is set = TRUE. If FALSE is input, these options will not be used.

The next card contains an * in the first column.

The next card is either a new control card for the next case of data or a blank card so that the run is terminated under normal circumstances.

Appendix B
SAMPLE CASE

B.1 DESCRIPTION

The output from a sample case is presented in this Appendix. Data are synthesized in order to test logical sequences. No significance should be attached to the values used. The listing includes a module map so that storage requirements are defined for each subroutine and common block. The program first prints out input data for easy reference and to provide a check on punched data. If probabilistic data are input, then two lines of output are provided for each item; the first line represents the most likely values input while the subsequent line represents the expected values calculated by the program.

Each section of output is described in detail:

(1) STAGE COST DATA include stage title or identifying name, recurring cost of first unit and learning curve (LC) factor for hardware, ETR, and WTR recurring cost, respectively. If any stage has jump – discontinuous form of recurring cost for any of the above three types – then the following line provides relevant information. Development and sustaining cost for each stage are listed along with years of availability referenced to the initial launch year. Each stage may belong to at most four "shared cost groups," whose reference numbers are listed on the output. Each group number is referenced to the "Shared Cost Data" number which follow this section. "Batch Fact" defines the number of years over which vehicles may be considered as produced in one period of time for learning curve purposes. A a reusable stage is designated as such and its expected unit purchase price is given with the input most likely value in parenthesis. Any miscellaneous costs associated with a stage are output as fixed costs. The annual expenditure is provided over a 12 year period.

(2) SHARED COST DATA include data on each shared cost group which was referenced in (1) above. These groups may be families such as the Titan family or they may be subsystems, such as a guidance system shared by several stages. Total development cost for any vehicle equals the sum of the development costs for each of its component stages plus any development costs for any shared groups associated with these stages plus any integration development costs required. As mentioned above, the first line represents the most likely value while the second line (if appropriate) represents the expected value calculated by the program.

(3) INTEGRATION COST DATA are always between "families." If a specific stage-to-stage integration cost is desired, each stage must be put in a shared cost group by itself. Thus, many shared cost groups in (2) above will have no associated non-recurring costs. These groups will, however, be integrated with other shared cost groups, and this combination does have an integration non-recurring cost.

(4) PAD COST DATA would normally be the section which follows. For simplicity, no pad costs were included with this test run, but this section would list the complex reference number, identifying name and location, e.g. TITE represents Titan ETR complex, and the next entry would show the maximum number of launches per year per pad at this complex. All possible combinations of pad-related costs are listed with their respective values for each pad.

(5) MISSION MODEL DATA include mission internal reference number, identifying name, total ΔV required, payload required in lb, priority value, launch site identification, (1 = ETR, 2 = WTR), and launch rate schedule by year.

The following page lists all most likely costs (modal) associated with each mission and then lists all the corresponding calculated expected costs. Payload recurring costs (PLR) are distributed over a 4-year period by the following four fractions, where the last year is the year of launch. Development

costs (DEV) are included along with the development period and starting year. Sustaining costs (SUST) and total miscellaneous fixed costs (FIXED) are included for future reference.

(6) SPECIAL PROGRAMS are listed by internal reference number and name. Development cost (DEV), start year and duration are included along with sustaining costs (SUST). Fixed costs, if any are input, are output by year. Both modal (input) and expected (calculated) values are output.

(7) INPUT DATA TOTALS include total number of each input item along with other pertinent information from control card.

(8) QUANTITIES BRANCHED UPON lists every non-zero, non-recurring cost or "budget option" which the algorithm will consider in the optimization process along with its availability status. The reference number listed is used in the optional logic output described in (11).

(9) VEHICLE/MISSION CAPABILITY is a matrix of final vehicle-to-mission compatibility presenting the results of subroutines CAPBLI and AVAILI. Each vehicle is listed by stage components and internal reference number. The vehicle/mission number on the top line represents the mission-year combination number (NM) while the mission number only is given on the following line at the top of the matrix.

(10) CHANGED QUANTITIES BRANCHED UPON is a section included only if reusable stages appear in the input. The number of units purchased is indicated and multiplied by the unit purchase price in order to determine the estimated investment cost for each reusable stage for that iteration. This investment cost is added to the actual development cost for use by the algorithm. In general, these "budget option" quantities are the only ones from the list in (8) that will vary from iteration to iteration.

(11) BRANCH AND BOUND NODE VALUES present optional information which enables the user to check the internal logic of the algorithm. Each node is given a reference number which it keeps until its associated total bound exceeds the value of a known solution. (****represents a very large number, denoting an unfeasible combination). The node number from which branching

is taking place is provided in the second column. The last new node to be generated at each branch is given the reference number of its parent node for continuity. The reference number of the cost item under consideration [see (8)] is listed in the next column along with the appropriate sustaining year for that node. (0 represents no development or sustaining for that cost item). The recurring, non-recurring, and total lower bounds are then provided so that each branch in the decision tree is represented.

When a final solution has been found, it is designated a POSSIBLE SOLUTION. If it is identically the same as a previous possible solution, this fact is printed out and the newly found solution is discarded. Otherwise, if some pad costs and small sustaining costs were ignored by the algorithm, these extra costs are computed and added to the lower bound of the corresponding node. The values of these costs are printed out below the node information for the possible solution under consideration. When the optimal solution has been found – the least cost possible solution already investigated – this fact is designated on the following page.

(12) SOLUTION NUMBER 1 – the optimal launch vehicle for each mission-year combination is printed out as well as an array of mission information for easy reference. The "Number of Launches" represents the launch rate by year multiplied by the priority factor and the number of trips required by the associated vehicle to satisfy the mission payload requirements.

Following the first solution is a description of the uncertainties associated with the total cost of this assignment. The lognormal parameters μ and σ^2 are output along with the lognormal densities at selected points. The 50 percent uncertainty interval, with lower bound taken as most likely value (mode), is also output.

(13) Sections (11) and (12) are repeated until NSOL = 3 assignments have been found. The second and third assignments have added information output since the probability that those assignments cost more than preceding

assignments is output for various levels of correlation. The proper level of correlation is determined by the analyst since he can determine how much increase in technology is required by each program. Two programs requiring approximately the same technological advance will have a high degree of correlation.

(14) THE OPTIMUM SOLUTION HAS BEEN DETERMINED signifies the successful completion of the algorithm. If no significantly different second and third best solutions can be found, this fact will be output here and the program will continue using the optimal solution found.

(15) Following the above selection of an optimal assignment, any input to subroutine SMOOTH is automatically output as it appears on the data cards. The program constraints are then output – first the input constraints, then the calculated constraints. "Average" recurring cost data for each of the vehicles in the optimum assignment are calculated in VEHRC and output on the following page. Each vehicle is assigned a key number which is used internally and output with the associated stage component names defining the vehicle.

(16) The breakdown of costs by program and type, and by program and year on the following pages, is essentially the same as for the original budget smoothing model. For example, Program 2 (PN = Program Number) has NAME MAPLSU for Manned Planetary Support. The development start date is 1984. The program has no development (DEVL) costs and hence no development duration (YRS). Sustaining costs (SUST) start in year 1984 (= START + SS - 1.). They are spent for 0 (SD) years. Recurring costs start in year 1987 (= START + RS - 1.) and last for 4 (RD) years. The distribution follows on the same line (e.g., $133 M in 1987, $405M in 1988, $438M in 1989, and $410M in 1990). On the following line fixed miscellaneous costs are similarly listed if any have been input for that program [e.g., fixed costs start in year 1971 (= START + RS - 1.) and last for 5 years (RD) for Program Number 13]. The distribution follows on the same line of output.

More complete data on these entries are provided in reference 1. Programs associated with missions are output first. For the selected sample case Programs 1 through 12 are mission related. Program 13 is a miscellaneous program having no associated launches, and the remaining programs are development or sustaining costs associated with launch vehicles. (There are only two such programs for this test case.) These last programs are identified by the decision number used in the ASSIGN algorithm. A list of decision numbers, their associated values, and types of expenditure has been output previously for reference.

(17) The section "Total Program Costs and Launch Vehicle Schedule" is output as in the original smoothing program with the following modification. Instead of printing the launch vehicle key name under its associated program and year of launch, the key number already output with each corresponding vehicle name is substituted for simplicity. Total program costs are output by year as they would actually be spent.

(18) A plot follows this tabulated data showing actual yearly totals (*) and desired yearly level of spending (0). The modal value is plotted as an M while the upper value of the 50 percent uncertainty interval is designated by a U. Fixed costs are plotted by an F. Normally under options MOS = 0 or 1, the smoothed data are then output using the same formats. Only data input to SMOOTH directly from ASSIGN and the final smoothed data are output. Intermediate output is supressed. For this sample run, MOS = 2 was specified so no smoothing was performed. If no new case data are input, then the normal termination of the run is designated by END OF DATA – JOB COMPLETE.

Any discrepancies in input data are noted and printed out as a warning to the user. The flow diagrams in Appendix C define all non-normal exits from the algorithm in CHOOZS. Each non-normal termination of a case is denoted by a printout of the qualifying reason. The program then reads in new case data, if available, and proceeds as normal.

The sample case included in this Appendix required 0.52 minutes on the 360/67 computer available at Ames Research Center, Moffett Field, California. Estimating run time is quite difficult for a new set of data since the number of solutions "close to" the optimum solution determines how large the decision tree will be and, as a consequence, how much computer time must be expended. As a general rule, the computer time increases linearly with the number of missions in the mission model and exponentially with the number of decision items determined by the stage, shared group, integration, and pad cost input.

B.2 SAMPLE CASE PRINTOUT

The computer printout for the sample case discussed above follows:

```
//MOX02BB  JOB  (T3582,TEST,1,1),'GOLDEN'          STOP 4
IEF236I ALLOC. FOR MOX02BB  LKED
IEF237I SYSLIB   ON 1C0
IEF237I SYSLMOD  ON 333
IEF237I DECKS    ON 336
IEF237I          ON 330
IEF237I          ON 330
IEF237I SYSPRINT ON 0A1
IEF237I SYSUT1   ON 335
IEF237I SYSLIN   ON 051
IEF285I   SYS1.FORTLIB                                    KEPT
IEF285I   VOL SER NOS= SYSLB1.
IEF285I   SYS71312.T161623.RF000.MOX02BB.GOSET            PASSED
IEF285I   VOL SER NOS= 555555.
IEF285I   SYS1.USERLIB                                    KEPT
IEF285I   VOL SER NOS= USER01.
IEF285I   SYS1.USERLIB2                                   KEPT
IEF285I   VOL SER NOS= 222222.
IEF285I   SYS1.USERLIB3                                   KEPT
IEF285I   VOL SER NOS= 222222.
IEF285I   SYS71312.T161623.RF000.MOX02BB.R0000689         DELETED
IEF285I   VOL SER NOS=        .
IEF285I   SYS71312.T161623.RF000.MOX02BB.R0000690         DELETED
IEF285I   VOL SER NOS= USER02.
IEF285I   SYS71312.T161623.RF000.MOX02BB.R0000691         DELETED
IEF285I   VOL SER NOS=        .
IEF236I ALLOC. FOR MOX02BB  GO
IEF237I PGM=*.DD ON 333
IEF237I FT05F001 ON 063
IEF237I FT06F001 ON 0A1
IEF237I GOSET    ON 333
IEF285I   SYS71312.T161623.RF000.MOX02BB.GOSET            PASSED
IEF285I   VOL SER NOS= 555555.
IEF285I   SYS71312.T161623.RF000.MOX02BB.R0000693         DELETED
IEF285I   VOL SER NOS=        .
IEF285I   SYS71312.T161623.RF000.MOX02BB.R0000692         DELETED
IEF285I   VOL SER NOS=        .
IEF285I   SYS71312.T161623.RF000.MOX02BB.GOSET            DELETED
IEF285I   VOL SER NOS= 555555.
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,OVLY,MAP
           VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      INCLUDE DECKS(MOX02MN,MOX01PK)
IEW0000      INCLUDE DECKS(MOX02NI,MOX02NR)
IEW0000      ENTRY MAIN
IEW0000        OVERLAY A
IEW0000      INCLUDE DECKS(MOX02AS)
IEW0000        OVERLAY B
IEW0000      INCLUDE DECKS(MOX02DS)
IEW0000        OVERLAY C
IEW0000      INCLUDE DECKS(MOX02EX)
IEW0000      OVERLAY B
IEW0000      INCLUDE DECKS(MOX02CI)
IEW0000        OVERLAY C
IEW0000      INCLUDE DECKS(MOX02MI,MOX02PI,MOX02ME)
IEW0000        OVERLAY B
IEW0000      INCLUDE DECKS(MOX02DN,MOX02MH,MOX02PN,MOX02AL)
IEW0000        OVERLAY B
IEW0000      INCLUDE DECKS(MOX02SM,MOX02RS,MOX02VC)
IEW0000        OVERLAY B
IEW0000      INCLUDE DECKS(MOX02CM)
IEW0000        OVERLAY C
IEW0000      INCLUDE DECKS(MOX02LD,MOX02OI)
IEW0000        OVERLAY C
IEW0000      INCLUDE DECKS(MOX02CM)
IEW0000        OVERLAY C
IEW0000      INCLUDE DECKS(MOX02PC)
IEW0000        OVERLAY A
IEW0000      INCLUDE DECKS(MOX02SS,MOX02RV,MOX02CR,ALINPT,MOX01UP,MOX02AT)
IEW0000      INCLUDE DECKS(MOX02TC)
IEW0000      INCLUDE DECKS(MOX02SH,MOX02LC)
```

MODULE MAP

CONTROL SECTION				ENTRY					
NAME	ORIGIN	LENGTH	SEG. NO.	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
$SEGTAB	00	4C	1						
MAIN	50	1984	1						
PACK	19D8	E8	1						
				UNPACK	1A2A	ITEM	1A6E		
NOTRI	1AC0	2A6	1						

NDTR	1D68	1DC	1
IHCSLOG *	1F48	1BA	1
IHCSSCN *	2108	1ED	1
IHCSEXP *	22F8	1B0	1
IHCFRXPI*	24A8	141	1
IHCFRXPR*	25F0	183	1

ALOG10	1F48	ALOG	1F64
COS	2108	SIN	2124
EXP	22F8		
FRXPI=	24A8		

NAME	ORIGIN	LENGTH	SEG. NO.
IHCECOMH*	2778	F31	1
IHCCOMH2*	36B0	545	1
IHCFMAXI*	3BF8	C9	1
IHCSSQRT*	3CC8	149	1
IHCFCVTH*	3E18	1175	1
IHCEFNTH*	4F90	512	1
FIOCS= *	54A8	160	1
IHCEFIOS*	5608	111C	1
IHCERRM *	6728	5AC	1
IHCUOPT *	6CD8	398	1
IHCETRCH*	7070	28E	1
IHCUATBL*	7300	638	1
SAVER	7938	FC0	1
SAVEXP	88F8	14BC	1
SAVSAR	9DB8	A5C	1
SAVE1	A818	FC4	1
SAV2	B7E0	FE0	1
SAV3	C7C0	9E4	1
SAV4	D1A8	3188	1
SVACAV	10330	B48	1
SAVALL	10678	3A1C	1
VARACE	14898	ADC	1
SCRACH	15378	6A60	1
$ENTAB	1BDD8	24	1
ASSIGN	16E00	8DE	2
TEMP	1C6E0	4110	2
$ENTAB	207F0	54	2

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
FRXPR=	25F0						
IBCOM=	2778	FDIOCS=	2834	INTSWTCH	3696		
SEQDASD	3910						
MAXO	3BF8	MINO	3C0E	AMAXO	3C24	AMINO	3C3A
SQRT	3CC8						
ADCON=	3E18	FCVAOUTP	3EC2	FCVLOUTP	3F52	FCVZOUTP	40A2
FCVIOUTP	442E	FCVEOUTP	4930	FCVCOUTP	484A	INT6SWCH	4E33
ARITH=	4F90	ADJSWTCH	52FC				
SETB99	5534	RESB99	554E				
FIOCSBEP	560E						
ERRMON	6728	IHCERRE	6740				
IHCTRCH	7070	ERRTRA	7078				

DATINS	20848	3ED0	3
$ENTAB	24718	18	3
MEAN	24730	23A	4
CAPBLI	20848	B40	5
$ENTAB	21388	18	5

NAME	ORIGIN	LENGTH	SEG. NO.
MISMTI	213A0	466	6
PERFI	21808	4BC	6
MATEI	21CC8	A68	6
DECSNI	20848	11FE	7
MATCHI	21A48	1984	7
PRINTI	23300	108A	7
AVAILI	24460	7A8	7
STGNMI	20848	2314	8
REUSE	22860	4D8	8
VEHRC	23038	3D2	8
CHOOZS	20848	2280	9
$ENTAB	22AC8	3C	9
LBONDI	22808	A50	10
OUTPTI	23558	58C	10
CMPARE	22808	2870	11
PDCSTI	22808	1BC2	12
SMOOTH	1BE00	22AC	13
REVLUS	1E080	672	13
CONSTR	1E728	4F2	13
ALINPT	1EC20	B1A	13
CMPLOT	1F740	F68	13
AFRMT	206A8	40	13
TCOSTS	206E8	1A46	13
SHIFTS	22130	7C2	13

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
INPUT	1EC20						
PLOT1	1F772	PLOT2	1F98E	PLOT3	1FB52	PLOT4	1FCA2
CMIT	1FF06	PLTAPE	1FF3C				

```
LISTC       228F8      784     13
SAVRT       23080      FA0     13

ENTRY ADDRESS          50
TOTAL LENGTH        25678

****MAIN        DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
```

STAGE COST DATA

TITLE	RECURRING LC (HARDWARE)		RECURRING LC (ETR ONLY)		RECURRING LC (WTR ONLY)		DEVELOPMENT	SUSTAINING	AVAILABLE FROM	AVAILABLE TO	SHARED COST GROUPS				BATCH FACT
S-1B	38.00	0.0	0.0	0.0	0.0	0.0	95.00	90.00	4	20	0	0	0	0	1
S-1B	39.36	0.0	0.0	0.0	0.0	0.0	97.90	92.98	4	20	0	0	0	0	1
S-1C	55.00	0.0	0.0	0.0	0.0	0.0	110.00	147.00	4	20	14	17	0	0	1
S-1C	57.08	0.0	0.0	0.0	0.0	0.0	113.65	152.03	4	20	14	17	0	0	1
S-II	41.00	0.0	0.0	0.0	0.0	0.0	0.0	90.00	4	20	14	0	0	0	1
S-II	42.31	0.0	0.0	0.0	0.0	0.0	0.0	92.98	4	20	14	0	0	0	1
FIXED COSTS =	65.00		65.00	0.0	0.0		0.0	0.0	0.0		0.0		0.0		0.0
FIXED COSTS =	67.16		67.16	0.0	0.0		0.0	0.0	0.0		0.0		0.0		0.0
S-4B	21.00	0.0	0.0	0.0	0.0	0.0	60.00	65.00	4	20	14	0	0	0	1
S-4B	21.65	0.0	0.0	0.0	0.0	0.0	61.99	67.41	4	20	14	0	0	0	1
LS4B	14.30	0.0	0.0	0.0	0.0	0.0	45.00	15.00	5	20	13	0	0	0	1
LS4B	19.03	0.0	0.0	0.0	0.0	0.0	60.48	19.90	5	20	13	0	0	0	1
1200	27.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	20	12	18	20	0	1
1200	30.64	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	20	12	18	20	0	1
1565	26.50	0.0	0.0	0.0	0.0	0.0	220.00	20.00	5	20	11	0	0	0	1
1565	35.44	0.0	0.0	0.0	0.0	0.0	295.49	26.79	5	20	11	0	0	0	1
R25B	3.39	0.0	0.0	0.0	0.0	0.0	3199.00	244.70	8	20	0	0	0	0	1
R25B	6.98	0.0	0.0	0.0	0.0	0.0	6560.76	502.06	8	20	0	0	0	0	1
REUSABLE STAGE	UNIT PURCHASE PRICE= 347.18 (169.40)														
R25D	2.31	0.0	0.0	0.0	0.0	0.0	3739.00	178.20	8	20	21	0	0	0	1
R25D	4.77	0.0	0.0	0.0	0.0	0.0	7667.82	365.04	8	20	21	0	0	0	1
REUSABLE STAGE	UNIT PURCHASE PRICE= 238.27 (116.10)														
R1.5	6.60	0.0	0.0	0.0	0.0	0.0	4578.00	280.00	8	20	0	0	0	0	1
R1.5	13.62	0.0	0.0	0.0	0.0	0.0	9388.42	574.39	8	20	0	0	0	0	1
REUSABLE STAGE	UNIT PURCHASE PRICE= 286.76 (140.00)														

FIXED COSTS =		10.00		10.00		0.0		0.0		0.0	0.0	0.0		0.0		0.0		0.0		0.0		0.0
FIXED COSTS =		13.02		13.02		0.0		0.0		0.0	0.0	0.0		0.0		0.0		0.0		0.0		0.0
SSTO	2.42	0.0		0.0	0.0		0.0	0.0			3000.00	284.60	8	20		0	0	0	0		1	
SSTO	4.76	0.0		0.0	0.0		0.0	0.0			5864.57	556.72	8	20		0	0	0	0		1	
REUSABLE STAGE			UNIT PURCHASE PRICE= 281.33 (144.00)																			
S/C	2.00	0.0		0.0	0.0		0.0	0.0			1900.00	110.00	5	20		0	0	0	0		1	
S/C	2.60	0.0		0.0	0.0		0.0	0.0			2473.46	143.20	5	20		0	0	0	0		1	
REUSABLE STAGE			UNIT PURCHASE PRICE= 111.03 (85.00)																			
CSM	40.00	0.0		0.0	0.0		0.0	0.0			150.00	85.00	5	20		0	0	0	0		1	
CSM	41.33	0.0		0.0	0.0		0.0	0.0			154.97	88.08	5	20		0	0	0	0		1	

SHARED COST DATA

NO.	TITLE	DEVELOPMENT	SUSTAINING
14	SATN	0.0	110.00
14	SATN	0.0	113.65
17	SIVB	0.0	0.0
11	156	1.00	2.00
11	156	1.24	2.20
12	120	18.00	23.00
12	120	20.22	26.00
13	LS4B	0.0	0.0
18	1205	47.00	0.0
18	1205	52.96	0.0
20	1200	60.00	0.0
20	1200	67.40	0.0
21	R25U	0.0	0.0

INTEGRATION COST DATA

LOWER GROUP	UPPER GROUP	RECURRING	LC	DEVELOPMENT	SUSTAINING
156	LS4B	0.10	0.0	80.00	1.00
156	LS4B	0.12	0.0	107.17	1.24
120	LS4B	0.0	0.0	80.00	0.0
120	LS4B	0.0	0.0	107.17	0.0
156	R25U	0.0	0.0	50.00	0.0
156	R25U	0.0	0.0	84.99	0.0

MISSION MODEL

	MISSION	VELOCITY	PAYLOAD	PRIORITY	TR	LAUNCH SCHEDULE																	
						71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
1	MANPLA	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	
2	MAPLSU	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
3	MANLUN	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0
4	MALUSU	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	12	12	12
5	SPBASE	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
6	SPBASU	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	12	12	12	12	12	12	12
7	MEDO	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
8	MEOSUP	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10	10	10
9	MEDO	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0
10	MEOSUP	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	0	8	8	8	0	0	0	0	0	0
11	MEDO	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0
12	MEOSUP	29000.	25000.	1.00	1	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0

1	MANPLA	PLR= 90.0	DIST BY.15, .35, .25, .25,	DEV= 17500.0	FOR	6 YRS	STARTING	1984	SUST=	800.0	FIXED=	0.0
1	MANPLA	PLR= 153.0	DIST BY.15, .35, .25, .25,	DEV= 29748.0	FOR	6 YRS	STARTING	1984	SUST=	1359.9	FIXED=	0.0
2	MAPLSU	PLR= 50.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1984	SUST=	0.0	FIXED=	10.0
2	MAPLSU	PLR= 85.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1984	SUST=	0.0	FIXED=	11.0
3	MANLUN	PLR= 90.0	DIST BY.15, .35, .25, .25,	DEV= 17500.0	FOR	6 YRS	STARTING	1980	SUST=	800.0	FIXED=	0.0
3	MANLUN	PLR= 153.0	DIST BY.15, .35, .25, .25,	DEV= 29748.0	FOR	6 YRS	STARTING	1980	SUST=	1359.9	FIXED=	0.0
4	MALUSU	PLR= 50.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1980	SUST=	0.0	FIXED=	0.0
4	MALUSU	PLR= 85.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1980	SUST=	0.0	FIXED=	0.0
5	SPBASE	PLR= 80.0	DIST BY.15, .35, .25, .25,	DEV= 10000.0	FOR	7 YRS	STARTING	1977	SUST=	550.0	FIXED=	0.0
5	SPBASE	PLR= 104.1	DIST BY.15, .35, .25, .25,	DEV= 13018.2	FOR	7 YRS	STARTING	1977	SUST=	716.0	FIXED=	0.0
6	SPBASU	PLR= 45.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1977	SUST=	0.0	FIXED=	0.0
6	SPBASU	PLR= 76.5	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1977	SUST=	0.0	FIXED=	0.0
7	MEDO	PLR= 85.0	DIST BY.15, .35, .25, .25,	DEV= 3690.0	FOR	7 YRS	STARTING	1979	SUST=	217.3	FIXED=	0.0
7	MEDO	PLR= 161.4	DIST BY.15, .35, .25, .25,	DEV= 4803.7	FOR	7 YRS	STARTING	1979	SUST=	282.9	FIXED=	0.0
8	MEOSUP	PLR= 40.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1979	SUST=	0.0	FIXED=	0.0
8	MEOSUP	PLR= 52.1	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1979	SUST=	0.0	FIXED=	0.0
9	MEDO	PLR= 85.0	DIST BY.15, .35, .25, .25,	DEV= 3690.0	FOR	7 YRS	STARTING	1975	SUST=	217.3	FIXED=	0.0
9	MEDO	PLR= 161.4	DIST BY.15, .35, .25, .25,	DEV= 4803.7	FOR	7 YRS	STARTING	1975	SUST=	282.9	FIXED=	0.0
10	MEOSUP	PLR= 40.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1975	SUST=	0.0	FIXED=	0.0
10	MEOSUP	PLR= 52.1	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1975	SUST=	0.0	FIXED=	0.0
11	MEDO	PLR= 85.0	DIST BY.15, .35, .25, .25,	DEV= 3690.0	FOR	7 YRS	STARTING	1973	SUST=	217.3	FIXED=	0.0
11	MEDO	PLR= 161.4	DIST BY.15, .35, .25, .25,	DEV= 4803.7	FOR	7 YRS	STARTING	1973	SUST=	282.9	FIXED=	0.0
12	MEOSUP	PLR= 40.0	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1973	SUST=	0.0	FIXED=	0.0
12	MEOSUP	PLR= 52.1	DIST BY.10, .30, .30, .30,	DEV= 0.0	FOR	0 YRS	STARTING	1973	SUST=	0.0	FIXED=	0.0

SPECIAL PROGRAMS

1 PLANED	DEV = 0.0	SUST = 0.0	DEV STARTS 1971	FOR	0 YEARS							
FIXED COSTS =	1380.00	1400.00	800.00	41.00	1.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FIXED COSTS =	1425.73	1446.39	826.51	42.36	1.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0

NUMBER OF STAGES	13
NUMBER OF VEHICLES	8
NUMBER OF FAMILIES	8
NUMBER OF INTEGRATION COSTS	3
NUMBER OF PAD COMPLEXES	0
NUMBER OF MISSIONS	12
NUMBER OF YEARS	20
LAUNCH BASE YEAR	71
TOTAL ESTIMATE	300000.00
OPTION NUMBER	3
NUMBER OF SOLUTIONS	3
INFLATION FACTOR	.0
CORRELATION	.50

QUANTITIES BRANCHED UPON

NUMBER	DEVELOPMENT	SUSTAINING		YEAR AVAIL	LAST YEAR	DEV START	DEV DURATION
1	97.90	92.98	S-IB STAGE HARDWARE	4	20	1974	1.
2	113.65	152.03	S-IC STAGE HARDWARE	4	20	1974	1.
3	134.31	92.98	S-II STAGE HARDWARE	4	20	1974	1.
4	61.99	67.41	S-4B STAGE HARDWARE	4	20	1974	1.
5	60.48	19.90	LS4B STAGE HARDWARE	5	20	1973	3.
6	295.49	26.79	1565 STAGE HARDWARE	5	20	1973	3.
7	6560.76	502.06	R25B STAGE HARDWARE	8	20	1972	7.
8	7667.62	365.04	R250 STAGE HARDWARE	8	20	1972	7.
9	9414.45	574.39	R1.5 STAGE HARDWARE	8	20	1972	7.
10	5884.57	556.72	SSTO STAGE HARDWARE	8	20	1972	7.
11	2473.46	143.20	S/C STAGE HARDWARE	5	20	1973	3.
12	154.97	88.08	CSM STAGE HARDWARE	5	20	1974	2.
13	0.0	113.65	SATN SHARED HARDWARE	4	20	1971	0.
14	1.24	2.20	156 SHARED HARDWARE	3	20	1971	3.
15	20.22	26.00	120 SHARED HARDWARE	1	20	1971	3.
16	52.96	0.0	1205 SHARED HARDWARE	1	20	1971	3.
17	67.40	0.0	1200 SHARED HARDWARE	1	20	1971	3.
18	107.17	1.24	INTEGRATION OF 156 AND LS4B HARDWARE	3	20	1971	3.
19	107.17	0.0	INTEGRATION OF 120 AND LS4B HARDWARE	3	20	1971	3.
20	84.99	0.0	INTEGRATION OF 156 AND R25D HARDWARE	3	20	1971	3.

VEHICLE / MISSION CAPABILITY
(1 = POSSIBLE, 0 = IMPOSSIBLE)

VEHICLE / MISSION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
MISSION NUMBER	1	2	3	4	4	4	4	4	5	6	6	6	6	6	6	6	7	8	8	8	8	8	9	10	10	10	11	12																			
1 S-IB S-4B CSM	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1																			
2 S-IC S-II S-4B CSM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			
3 1200 LS4B CSM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			
4 1565 LS4B S/C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			
5 1565 R250	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			
6 R1.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			
7 SSTO S/C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			
8 R25B R250	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																			

CHANGED QUANTITIES BRANCHED UPON

NUMBER	DEVELOPMENT	SUSTAINING		YEAR AVAIL	LAST YEAR
7	8643.82	502.06	R25B STAGE HARDWARE	8	20
	NUMBER OF UNITS PURCHASED = 6.0				
8	9097.43	365.04	R25O STAGE HARDWARE	8	20
	NUMBER OF UNITS PURCHASED = 6.0				
9	11134.98	574.39	R1.5 STAGE HARDWARE	8	20
	NUMBER OF UNITS PURCHASED = 6.0				
10	7552.52	556.72	SSTO STAGE HARDWARE	8	20
	NUMBER OF UNITS PURCHASED = 6.0				
11	3139.64	143.20	S/C STAGE HARDWARE	5	20
	NUMBER OF UNITS PURCHASED = 6.0				

B R A N C H A N D B O U N D N O D E V A L U E S

NODE NO.	BRANCHED FROM	COST NO.	YEARS SUSTAIN	RECURRING BOUND	NON-RECURRING BOUND	TOTAL BOUND
2	1	10	0	3335.48	531.91	3867.39
3	1	10	8	3335.48	8641.14	11976.63
4	1	10	10	3278.53	9730.22	13008.75
5	1	10	12	3212.82	10815.56	14028.38
6	1	10	14	3055.11	11861.57	14916.68
7	1	10	16	2787.88	13110.56	15898.44
8	1	10	18	2489.99	14521.88	17011.87
1	1	10	20	2091.33	16033.97	18125.30
9	2	8	0	3867.40	12283.76	16151.16
10	2	8	8	3867.40	21746.23	25613.63
11	2	8	10	3843.05	22028.48	25871.53
12	2	8	12	3814.95	22132.88	25947.84
13	2	8	14	3747.53	21361.33	25108.86
14	2	8	16	3633.28	19546.98	23180.26
15	2	8	18	3505.92	17440.63	20946.55
2	2	8	20	3335.48	14374.88	17710.36
16	3	8	0	3867.40	20393.00	24260.39
17	3	8	8	3867.40	29855.47	33722.86
18	3	8	10	3843.05	30137.71	33980.76
19	3	8	12	3814.95	30242.12	34057.07
20	3	8	14	3747.53	29470.57	33218.09
21	3	8	16	3633.28	27656.21	31289.49
22	3	8	18	3505.92	25549.87	29055.79
3	3	8	20	3335.48	22484.11	25819.59
23	4	8	0	3786.10	21115.54	24901.63
24	4	8	8	3786.10	30578.01	34364.11
25	4	8	10	3786.10	31283.75	35069.84
26	4	8	12	3758.00	31360.05	35118.05
27	4	8	14	3690.58	30559.64	34250.22
28	4	8	16	3576.33	28745.30	32321.63
29	4	8	18	3448.97	26638.95	30087.92
4	4	8	20	3278.53	23573.19	26851.72
30	5	8	0	3692.29	21669.00	25361.29
31	5	8	8	3692.29	31131.48	34823.77
32	5	8	10	3692.29	31837.21	35529.50
33	5	8	12	3692.29	32539.20	36231.49
34	5	8	14	3624.87	31700.22	35325.09
35	5	8	16	3510.62	29830.64	33341.25
36	5	8	18	3383.26	27724.29	31107.55
5	5	8	20	3212.82	24658.53	27871.35

```
 37        6        8        0        3467.16        21438.51        24905.66
 38        6        8        8        3467.16        30900.98        34368.14
 39        6        8       10        3467.16        31606.71        35073.87
 40        6        8       12        3467.16        32308.70        35775.86
 41        6        8       14        3467.16        32971.36        36438.52
 42        6        8       16        3352.91        31042.77        34395.67
 43        6        8       18        3225.55        28809.06        32034.61
  6        6        8       20        3055.11        25704.54        28759.64
 44        7        8        0        3085.68        20274.73        23360.41
 45        7        8        8        3085.68        29737.21        32822.88
 46        7        8       10        3085.68        30442.94        33528.62
 47        7        8       12        3085.68        31144.93        34230.61
 48        7        8       14        3085.68        31807.59        34893.26
 49        7        8       16        3085.68        32423.43        35509.10
 50        7        8       18        2958.32        30189.72        33148.04
  7        7        8       20        2787.88        26953.53        29741.41
 51        9        9        0       16245.60         3604.02        19849.62
 52        9        9        8       16245.60        15313.39        31558.99
 53        9        9       10       15678.99        16462.17        32141.16
 54        9        9       12       15025.21        17610.95        32636.16
 55        9        9       14       13456.14        18759.73        32215.86
 56        9        9       16       10797.43        19908.51        30705.94
 57        9        9       18        7833.64        20529.04        28362.68
  9        9        9       20        3867.40        18602.05        22469.45
 58        8        8        0        2660.42        18849.63        21510.05
 59        8        8        8        2660.42        28312.11        30972.53
 60        8        8       10        2660.42        29017.84        31678.26
 61        8        8       12        2660.42        29719.83        32380.25
 62        8        8       14        2660.42        30382.48        33042.91
 63        8        8       16        2660.42        30998.32        33658.75
 64        8        8       18        2660.42        31601.05        34261.47
  8        8        8       20        2489.99        28364.85        30854.84
 65        2        7        0        3867.40        21395.21        25262.61
 66        2        7        8        3867.40        30541.09        34408.49
 67        2        7       10        3843.05        31199.52        35042.57
 68        2        7       12        3814.95        31804.76        35619.71
 69        2        7       14        3747.53        31851.57        35599.10
 70        2        7       16        3633.28        31233.55        34866.83
 71        2        7       18        3505.92        30429.38        33935.30
  2        2        7       20        3335.48        29013.59        32349.07
 72        1       11        0        3335.48        15321.73        18657.21
  1        1       11       20        2091.33        17929.46        20020.79
 73       72        8        0        3867.40        27073.59        30940.98
 74       72        8        8        3867.40        36536.06        40403.45
 75       72        8       10        3843.05        37290.49        41133.54
 76       72        8       12        3814.95        38048.66        41863.62
 77       72        8       14        3747.53        38846.17        42593.70
 78       72        8       16        3633.28        39690.50        43323.78
 79       72        8       18        3505.92        35306.22        38812.14
 72       72        8       20        3335.48        29164.70        32500.18
 80       51        5        0       45751.19          996.28        46747.46
 51       51        5       20       16245.60         3604.02        19849.62
 81       51       11        0       25844.74          463.21        26307.95
 51       51       11       20       16245.60         3604.02        19849.62
 82       51        6        0       25844.74         3602.85        29447.59
 51       51        6       20       16245.60         3604.02        19849.62
 83       51       18        0       25844.74         3898.34        29743.08
 51       51       18       20       16245.60         3604.02        19849.62
 84       51       14        0       25844.74         4005.51        29850.25
 51       51       14       20       16245.60         3604.02        19849.62

************ POSSIBLE SOLUTION    1  ************
 51                                  16245.60         3604.02        19849.62

EXTRA PAD COSTS =       0.0
    5     10      5      16.00      19.90     318.48
    6     10      5      16.00      26.79     747.17
   11     10      5      16.00     143.20    3038.37
   14     10      3      18.00       2.20    3077.95
   18     10      3      18.00       1.24    3100.24

EXTRA PAD & SMALL SUST COSTS =    3100.24
NEW VALUE =     22949.86

************ POSSIBLE SOLUTION    2  ************
  1                                   2091.33        17929.46        20020.79

EXTRA PAD COSTS =       0.0
   11     10      5      16.00     143.20    2291.20

EXTRA PAD & SMALL SUST COSTS =    2291.20
NEW VALUE =     22311.99
 80       15        9        0        7472.18        21682.46        29154.63
 85       15        9        8        7472.18        33391.83        40864.00
 86       15        9       10        7472.18        34194.91        41667.08
 87       15        9       12        7472.18        34944.80        42416.97
 88       15        9       14        7472.18        35136.28        42608.45
 89       15        9       16        7472.18        34662.91        42135.09
 90       15        9       18        7472.18        34003.40        41475.57
 15       15        9       20        3505.92        32076.41        35582.32
```

91	58	9	0	6626.68	17280.41	23907.09
92	58	9	8	6626.68	28989.78	35616.46
93	58	9	10	6626.68	30138.56	36765.24
94	58	9	12	6626.68	31287.34	37914.02
95	58	9	14	6626.68	32436.12	39062.80
96	58	9	16	6626.68	33584.90	40211.58
97	58	9	18	6626.68	34205.43	40832.11
58	58	9	20	2660.42	33485.41	36145.83

********** S O L U T I O N N U M B E R 1**********

1 RECURRING = 2091.33 NONRECURRING = 20220.66 TOTAL LAUNCH VEHICLE COST = 22311.99

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	RETURN PAYLOAD	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE	LAUNCH SITE
MANPLA	29000.	25000.	0.	1989	8.00	SSTO S/C	E
MAPLSU	29000.	25000.	10.	1990	15.00	SSTO S/C	E
MALUN	29000.	25000.	0.	1985	8.00	SSTO S/C	E
MALUSU	29000.	25000.	10.	1986	12.00	SSTO S/C	E
				1987	12.00	SSTO S/C	E
				1988	12.00	SSTO S/C	E
				1989	12.00	SSTO S/C	E
				1990	12.00	SSTO S/C	E
SPBASE	29000.	25000.	0.	1983	8.00	SSTO S/C	E
SPBASU	29000.	25000.	10.	1984	12.00	SSTO S/C	E
				1985	12.00	SSTO S/C	E
				1986	12.00	SSTO S/C	E
				1987	12.00	SSTO S/C	E
				1988	12.00	SSTO S/C	E
				1989	12.00	SSTO S/C	E
				1990	12.00	SSTO S/C	E
MEDD	29000.	25000.	0.	1985	7.00	SSTO S/C	E
MEDSUP	29000.	25000.	10.	1986	10.00	SSTO S/C	E
				1987	10.00	SSTO S/C	E
				1988	10.00	SSTO S/C	E
				1989	10.00	SSTO S/C	E
				1990	10.00	SSTO S/C	E
MEDD	29000.	25000.	0.	1981	7.00	SSTO S/C	E
MEDSUP	29000.	25000.	10.	1982	8.00	SSTO S/C	E
				1983	8.00	SSTO S/C	E
				1984	8.00	SSTO S/C	E
MEDD	29000.	25000.	0.	1979	7.00	SSTO S/C	E
MEDSUP	28000.	25000.	10.	1980	6.00	SSTO S/C	E

```
SOLUTION  1  HAS EXPECTED L V COST  22311.99  (  22311.99)     MODE =  16387.12   STD. DEV. =   9179.64
PARAMETERS MU AND SIGMASQ =      9.26   AND      0.41
 PROB (COST LE    16387. )  =0.18   DENSITY =    0.9386
PROB (COST LE    23567. )  =0.68    50 PERCENT UNCERTAINTY INTERVAL =    16387. TO    23567.   DENSITY =      0.51

 PROB (COST LE     9439. )  = .00   DENSITY =      .00
 PROB (COST LE    14047. )  =0.10   DENSITY =    0.7645
 PROB (COST LE    16929. )  =0.30   DENSITY =    0.9322
 PROB (COST LE    19920. )  =0.50   DENSITY =    0.7642
 PROB (COST LE    24105. )  =0.70   DENSITY =    0.4761
 PROB (COST LE    33280. )  =0.90   DENSITY =    0.1478
```

```
************ POSSIBLE SOLUTION   3 ************
  9                                3867.40       18602.05       22469.45

EXTRA PAD COSTS =       0.0

EXTRA PAD & SMALL SUST COSTS =      0.0
```

********** S O L U T I O N N U M B E R 2**********

9 RECURRING = 3867.40 NONRECURRING = 18602.05 TOTAL LAUNCH VEHICLE COST = 22469.45

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	RETURN PAYLOAD	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE	LAUNCH SITE
MANPLA	29000.	25000.	0.	1989	8.00	R1.5	E
MAPLSU	29000.	25000.	10.	1990	15.00	R1.5	E
MANLUN	29000.	25000.	0.	1985	8.00	R1.5	E
MALUSU	29000.	25000.	10.	1986	12.00	R1.5	E
				1987	12.00	R1.5	E
				1988	12.00	R1.5	E
				1989	12.00	R1.5	E
				1990	12.00	R1.5	E
SPBASE	29000.	25000.	0.	1983	8.00	R1.5	E
SPBASU	29000.	25000.	10.	1984	12.00	R1.5	E
				1985	12.00	R1.5	E
				1986	12.00	R1.5	E
				1987	12.00	R1.5	E
				1988	12.00	R1.5	E
				1989	12.00	R1.5	E
				1990	12.00	R1.5	E
MEOO	29000.	25000.	0.	1985	7.00	R1.5	E
MEOSUP	29000.	25000.	10.	1986	10.00	R1.5	E
				1987	10.00	R1.5	E
				1988	10.00	R1.5	E
				1989	10.00	R1.5	E
				1990	10.00	R1.5	E
MEOO	29000.	25000.	0.	1981	7.00	R1.5	E
MEOSUP	29000.	25000.	10.	1982	8.00	R1.5	E
				1983	8.00	R1.5	E
				1984	8.00	R1.5	E
MEOO	29000.	25000.	0.	1979	7.00	R1.5	E
MEOSUP	28000.	25000.	10.	1980	6.00	R1.5	E

SOLUTION 2 HAS EXPECTED L V COST 22469.45 (22469.45) MODE = 14059.72 STD. DEV. = 13004.52

PARAMETERS MU AND SIGMASQ = 9.13 AND 0.69

PROB (COST LE 14060.) =0.13 DENSITY = 0.9578

PROB (COST LE 21718.) =0.63 50 PERCENT UNCERTAINTY INTERVAL = 14060. TO 21718. DENSITY = 0.48

PROB (COST LE 9439.) = .00 DENSITY = .00

PROB (COST LE 12616.) =0.10 DENSITY = 0.8654

PROB (COST LE 15405.) =0.30 DENSITY = 0.9136

PROB (COST LE 18662.) =0.50 DENSITY = 0.6780

PROB (COST LE 23698.) =0.70 DENSITY = 0.3823

PROB (COST LE 36209.) =0.90 DENSITY = 0.1027

PROB (ASSIGNMENT 2 COST GE ASSIGNMENT 1 COST) =0.34 IF CORRELATION =0.0

PROB (ASSIGNMENT 2 COST GE ASSIGNMENT 1 COST) =0.33 IF CORRELATION =0.3

PROB (ASSIGNMENT 2 COST GE ASSIGNMENT 1 COST) =0.32 IF CORRELATION =0.6

PROB (ASSIGNMENT 2 COST GE ASSIGNMENT 1 COST) =0.27 IF CORRELATION =0.9

********** S O L U T I O N N U M B E R 3**********

51 RECURRING = 16245.60 NONRECURRING = 6704.26 TOTAL LAUNCH VEHICLE COST = 22949.86

MISSION TITLE	CHARACTERISTIC VELOCITY(FT/SEC)	PAYLOAD (LBS)	RETURN PAYLOAD	LAUNCH YEAR	NUMBER OF LAUNCHES	OPTIMUM LAUNCH VEHICLE	LAUNCH SITE
MANPLA	29000.	25000.	0.	1989	8.00	1565 LS4B S/C	E
MAPLSU	29000.	25000.	10.	1990	15.00	1565 LS4B S/C	E
MANLUN	29000.	25000.	0.	1985	8.00	1565 LS4B S/C	E
MALUSU	29000.	25000.	10.	1986	12.00	1565 LS4B S/C	E
				1987	12.00	1565 LS4B S/C	E
				1988	12.00	1565 LS4B S/C	E
				1989	12.00	1565 LS4B S/C	E
				1990	12.00	1565 LS4B S/C	E
SPBASE	29000.	25000.	0.	1983	8.00	1565 LS4B S/C	E
SPBASU	29000.	25000.	10.	1984	12.00	1565 LS4B S/C	E
				1985	12.00	1565 LS4B S/C	E
				1986	12.00	1565 LS4B S/C	E
				1987	12.00	1565 LS4B S/C	E
				1988	12.00	1565 LS4B S/C	E
				1989	12.00	1565 LS4B S/C	E
				1990	12.00	1565 LS4B S/C	E
MEDU	29000.	25000.	0.	1985	7.00	1565 LS4B S/C	E
MEDSUP	29000.	25000.	10.	1986	10.00	1565 LS4B S/C	E
				1987	10.00	1565 LS4B S/C	E
				1988	10.00	1565 LS4B S/C	E
				1989	10.00	1565 LS4B S/C	E
				1990	10.00	1565 LS4B S/C	E
MEDU	29000.	25000.	0.	1981	7.00	1565 LS4B S/C	E
MEDSUP	29000.	25000.	10.	1982	8.00	1565 LS4B S/C	E
				1983	8.00	1565 LS4B S/C	E
				1984	8.00	1565 LS4B S/C	E
MEDU	29000.	25000.	0.	1979	7.00	1565 LS4B S/C	E
MEDSUP	28000.	25000.	10.	1980	6.00	1565 LS4B S/C	E

SOLUTION 3 HAS EXPECTED L V COST 22949.86 (22949.92) MODE = 20632.64 STD. DEV. = 4939.20

PARAMETERS MU AND SIGMASQ = 9.45 AND 0.13

PROB (COST LE 20633.) =0.26 DENSITY = 1.2770

PROB (COST LE 25772.) =0.76 50 PERCENT UNCERTAINTY INTERVAL = 20633. TO 25772. DENSITY = 0.72

PROB (COST LE 9439.) = .00 DENSITY = .00

PROB (COST LE 17498.) =0.10 DENSITY = 0.8305

PROB (COST LE 19979.) =0.30 DENSITY = 1.2587

PROB (COST LE 22129.) =0.50 DENSITY = 1.1994

PROB (COST LE 24716.) =0.70 DENSITY = 0.8684

PROB (COST LE 29419.) =0.90 DENSITY = 0.3350

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 1 COST) =0.71 IF CORRELATION =0.0

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 1 COST) =0.72 IF CORRELATION =0.3

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 1 COST) =0.74 IF CORRELATION =0.6

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 1 COST) =0.79 IF CORRELATION =0.9

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 2 COST) =0.74 IF CORRELATION =0.0

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 2 COST) =0.75 IF CORRELATION =0.3

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 2 COST) =0.77 IF CORRELATION =0.6

PROB (ASSIGNMENT 3 COST GE ASSIGNMENT 2 COST) =0.81 IF CORRELATION =0.9

THE OPTIMUM SOLUTION HAS BEEN DETERMINED

```
 TITLE = 'TEST CASE',     LEVEL = 3180.,3500.,3850.,4230.,4650.,5110.,
 5620.,6180.,6800.,7480.,8130.,8940.,9830.,10810.,11890.,13080.,14390.,
 15830.,17410.,19150.,
     ISTRT = 2,    IFIN = 20,
 MAXITR = 10,                    FIXED =1800.,1650.,1500.,1540.,1590.,
 1630.,1680.,1740.,1800.,1870.,1930.,2010.,2100.,2200.,2310.,2430.,
 2560.,2700.,2860.,3140.,
     PMAX = 15500.,   PMIN  = 1500.,
     NCSTR = 11,      NPROG = 101,73,74,71,72,75,76,80,81,84,85,
     KODE = 8,10X6,
    KPROG = 0,74,73,72,71,76,75,81,80,85,84,
     CS = 0.,0.,-1.,0.,-1.,0.,-1.,0.,-1.,0.,-1.,
*
```

35 CONSTRAINTS

KODE								
8	13 PLANED FIXED							
6	TARGET DATE	11 MEOO	NO LATER THAN	0.	YEARS AFTER	12	MEOSUP	
6	TARGET DATE	12 MEOSUP	NO LATER THAN	1.	YEARS AFTER	11	MEOO	
6	TARGET DATE	9 MEOO	NO LATER THAN	0.	YEARS AFTER	10	MEOSUP	
6	TARGET DATE	10 MEOSUP	NO LATER THAN	1.	YEARS AFTER	9	MEOO	
6	TARGET DATE	7 MEOO	NO LATER THAN	0.	YEARS AFTER	8	MEOSUP	
6	TARGET DATE	8 MEOSUP	NO LATER THAN	1.	YEARS AFTER	7	MEOO	
6	TARGET DATE	5 SPBASE	NO LATER THAN	0.	YEARS AFTER	6	SPBASU	
6	TARGET DATE	6 SPBASU	NO LATER THAN	1.	YEARS AFTER	5	SPBASE	
6	TARGET DATE	1 MANPLA	NO LATER THAN	0.	YEARS AFTER	2	MAPLSU	
6	TARGET DATE	2 MAPLSU	NO LATER THAN	1.	YEARS AFTER	1	MANPLA	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			1	MANPLA	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			1	MANPLA	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			2	MAPLSU	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			2	MAPLSU	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			3	MANLUN	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			3	MANLUN	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			4	MALUSU	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			4	MALUSU	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			5	SPBASE	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			5	SPBASE	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			6	SPBASU	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			6	SPBASU	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			7	MEOO	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			7	MEOO	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			8	MEOSUP	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			8	MEOSUP	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			9	MEOO	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			9	MEOO	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			10	MEOSUP	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			10	MEOSUP	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			11	MEOO	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			11	MEOO	
11	PROGRAM DEV	14	COMPLETED BY FIRST LAUNCH OF PROGRAM			12	MEOSUP	
11	PROGRAM DEV	15	COMPLETED BY FIRST LAUNCH OF PROGRAM			12	MEOSUP	

RECURRING COST DATA

KEY	NAME	UNIT COST
7	SSTOS/C	7.36

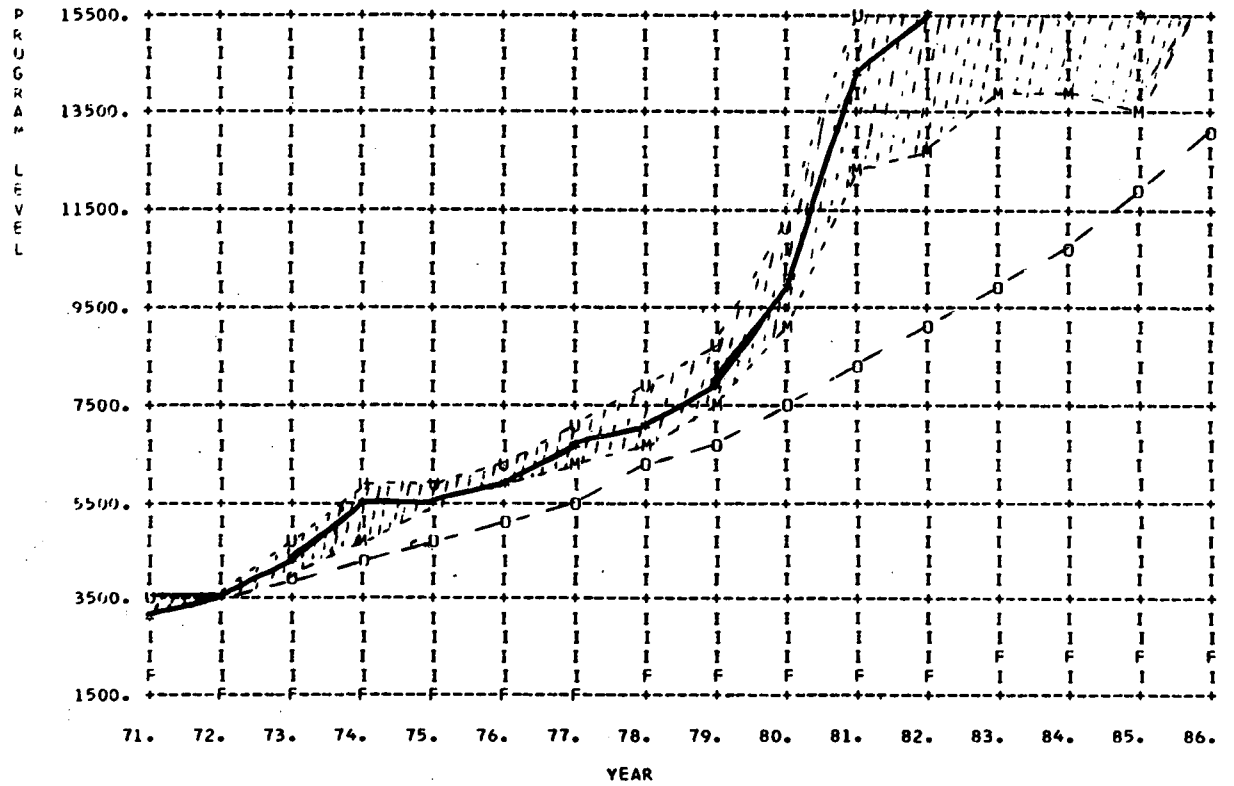
REFERENCE YEAR 1971. TEST CASE

PN	NAME	START	DEVL	YRS	SUST	SS	SD	RS	RD	RECURRING OR FIXED ITEMS										
1	MANPLA	1984.	29748.	6.	1360.	4	13	3	4	187.	440.	335.	321.							
2	MAPLSU	1984.	0.	0.	0.	1	0	4	4	133.	405.	438.	410.							
								4	4	1.	2.	3.	4.							
3	MANLUN	1980.	29748.	6.	1360.	4	13	3	4	187.	440.	335.	321.							
4	MALUSU	1980.	0.	0.	0.	1	0	4	8	106.	430.	780.	1108.	1108.	1002.	678.	328.			
5	SPRASE	1977.	13018.	7.	716.	5	13	4	4	128.	303.	238.	223.							
6	SPRASU	1977.	0.	0.	0.	1	0	5	10	96.	389.	709.	1006.	1006.	1006.	1006.	910.	617.	297.	
7	MEDU	1979.	4804.	7.	283.	5	8	4	4	172.	406.	308.	295.							
8	MEDSUP	1979.	0.	0.	0.	1	0	5	8	56.	227.	420.	594.	594.	539.	368.	175.			
9	MEDU	1975.	4804.	7.	283.	5	6	4	4	172.	406.	308.	295.							
10	MEDSUP	1975.	0.	0.	0.	1	0	5	6	45.	181.	336.	431.	294.	140.					
11	MEDU	1973.	4804.	7.	283.	5	5	4	4	172.	406.	308.	295.							
12	MEDSUP	1973.	0.	0.	0.	1	0	5	4	33.	103.	116.	105.							
13	PLANED	1971.	0.	0.	0.	1	0	0	0	0.										
								1	5	1426.	1446.	827.	42.	1.						
14	DEV 10	1972.	7553.	7.	557.	5	16	0	0											
15	DEV 11	1973.	3140.	3.	143.	2	18	0	0											
TOTAL			97617.		61526.															

TOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE

YEAR	1971.	1972.	1973.	1974.	1975.	1976.	1977.	1978.	1979.	1980.	1981.	1982.	1983.	1984.	1985.	1986.	1987.	1988.	1989.	1990.
PROGRAM																				
1 MANPLA	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1917.	5310.	7833.	9446.	7005.	3592. 7*	1360.
2 MAPLSU	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	134.	407.	441.	414. 7*
3 MANLUN	0.	0.	0.	0.	0.	0.	0.	0.	0.	1912.	5310.	7833.	9446.	7005.	3592. 7*	1360.	1360.	1360.	1360.	1360.
4 MALUSU	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	106.	430.	780.	1108. 7*	1108. 7*	1002. 7*	676. 7*	328. 7*
5 SPBASE	0.	0.	0.	0.	0.	0.	584.	1716.	2682.	3179.	3701.	2670.	1523. 7*	716.	716.	716.	716.	716.	716.	716.
6 SPBASU	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	96.	389.	709.	1006. 7*	1006. 7*	1006. 7*	1006. 7*	910. 7*	617. 7*	297. 7*
7 MEDU	0.	0.	0.	0.	0.	0.	0.	0.	215.	633.	990.	1298.	1678.	1224.	794. 7*	283.	283.	283.	283.	283.
8 MEDSUP	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	56.	227.	420.	594. 7*	594. 7*	539. 7*	368. 7*	175. 7*
9 MEDU	0.	0.	0.	0.	215.	633.	990.	1298.	1678.	1224.	794. 7*	283.	283.	283.						
10 MEDSUP	0.	0.	0.	0.	0.	0.	0.	0.	45.	181.	336.	431. 7*	294. 7*	140. 7*						
11 MEDU	0.	0.	215.	633.	990.	1298.	1678.	1224.	794. 7*	283.	283.									
12 MEDSUP	0.	0.	0.	0.	0.	0.	33.	103.	116.	105. 7*										
13 PLANED	1426.	1446.	827.	42.	1.															
14 DEV 10	0.	339.	996.	1556.	1770.	2112.	1552.	896.	557.	557.	557.	557.	557.	557.	557.	557.	557.	557.	557.	557.
15 DEV 11	0.	0.	828.	1615.	971.	143.	143.	143.	143.	143.	143.	143.	143.	143.	143.	143.	143.	143.	143.	143.
SUM	1426.	1785.	2866.	3846.	3947.	4187.	4981.	5380.	6230.	8217.	12209.	13604.	14796.	13643.	13318.	13601.	15348.	12921.	8755.	5632.
FIXED	1800.	1650.	1500.	1540.	1590.	1630.	1680.	1740.	1800.	1870.	1930.	2010.	2100.	2200.	2310.	2430.	2560.	2700.	2860.	3140.
TOTAL	3226.	3435. *	4366. *	5385. *	5537. *	5817. *	6661. *	7120. *	8030. *	10087. *	14139. *	15614. *	16896. *	15843. *	15628. *	16031. *	17908. *	15621. *	11615. *	8773. *
LEVEL	3180.	3500.	3850.	4230.	4650.	5110.	5620.	6180.	6800.	7480.	8130.	8940.	9830.	10810.	11890.	13080.	14390.	15830.	17410.	19150.
MODE	3186.	3320.	3906.	4584.	4655.	5022.	6156.	6715.	7500.	9269.	12147.	12591.	13919.	13792.	13536.	12928.	14881.	13564.	10726.	8045.
50 PER CENT CONFID.	3362.	3649.	4697.	5760.	5891.	6258.	7285.	7780.	8804.	11117.	15543.	16844.	18383.	17461.	17183.	17218.	19477.	17122.	12712.	9472.

RMS = 4264. SMOOTHING INTERVAL 1972. THRU 1990.



END OF DATA - JOB COMPLETE

JOB NO.	JOB TYPE	PROGRAMMER NAME	CP NO.	JOB ORDER	TIME (MIN) BEGIN EXEC	LINES PRINTED	JOB CPU MIN.	ACCT UNITS	PRIOR TYPE	DATE
A230	TEST	GOLDEN	MOX02B8	T3582	1097.38	903	0.52	0.14	N	11/08/71

Appendix C
FLOW CHARTS

C.1 DESCRIPTION

Flow charts are provided in this section for each of the major subroutines and the main program MASTER. They appear in alphabetical order by subroutine name. A short description of the purpose of each subroutine is provided in the program listing in Appendix D. Subroutines AFRMT, INPUT, PLOT, and PACK were written in 360 Assembler Language so a description of each subroutine appears in this section rather than a flow chart.

C.2 MAJOR SUBROUTINE CHARTS

The subroutine flow charts follow.

SUBROUTINE AFRMT

IDENTIFICATION

Subroutine AFRMT
Deck Name MOX02AT
Fortran IV subroutine coded in 360 Assembler Language
Written by R. E. Slye

PURPOSE

This subroutine converts a variable from integer to A format

USAGE

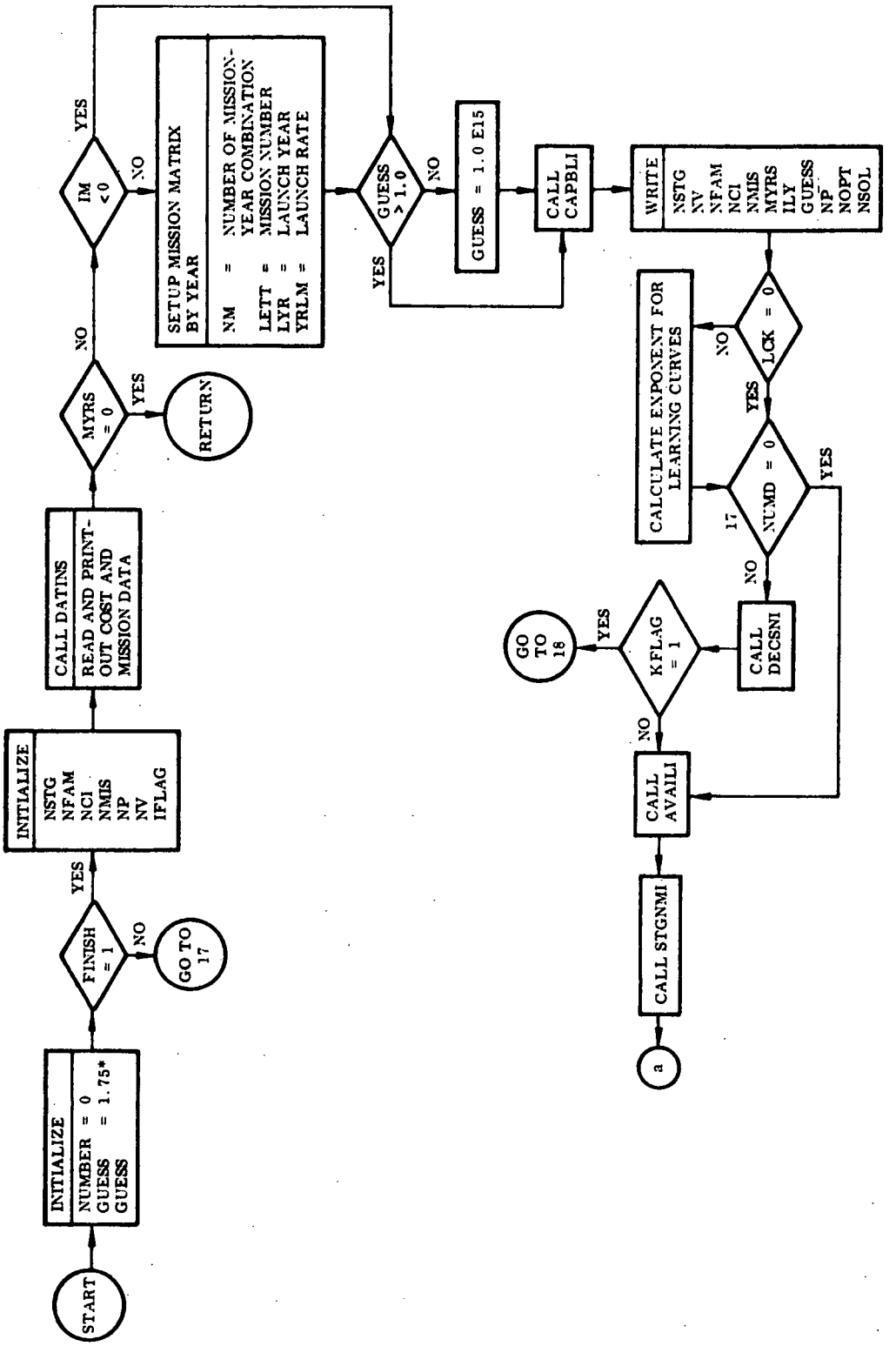
CALL AFRMT (I, X)

where

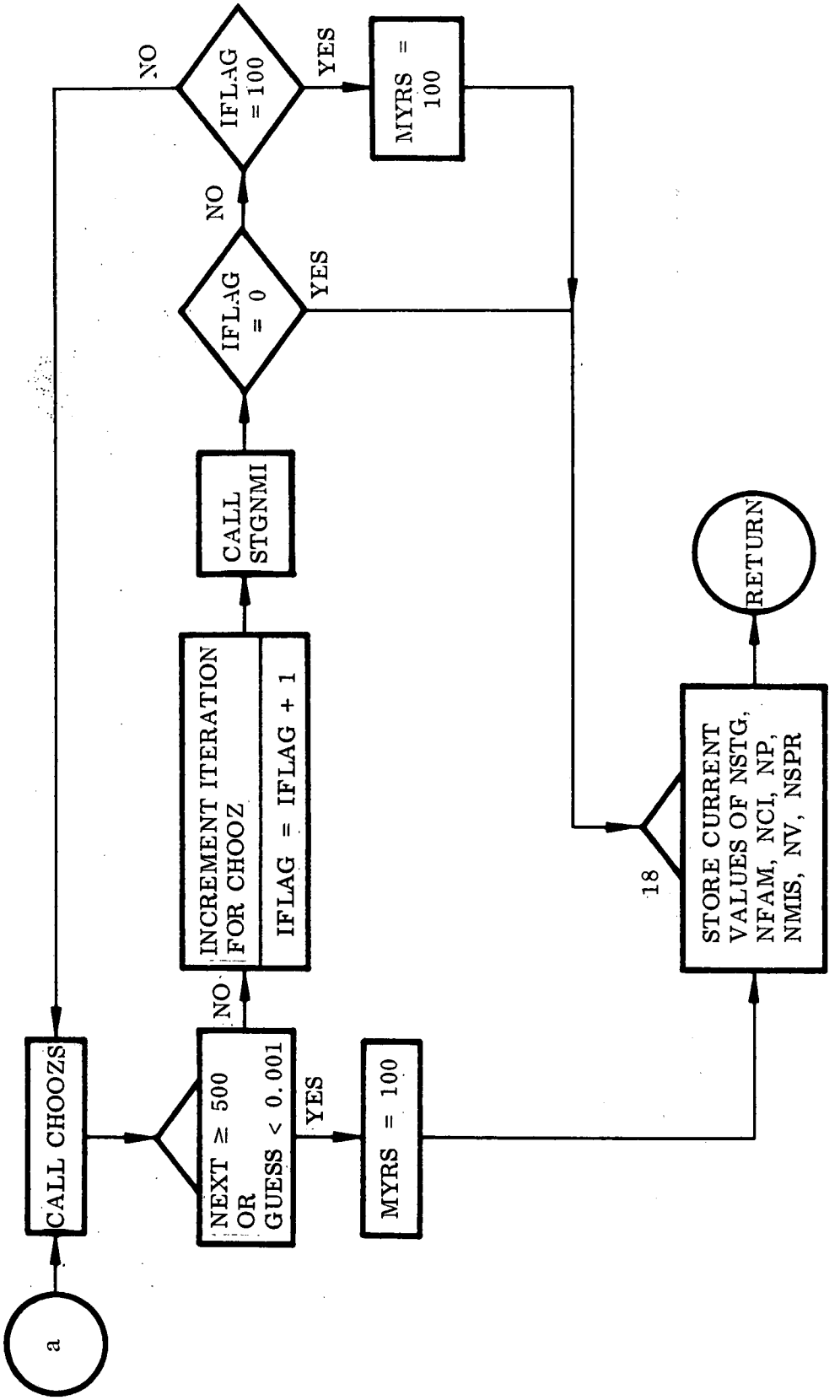
I is the name of the variable (may be one element of an array) in integer format

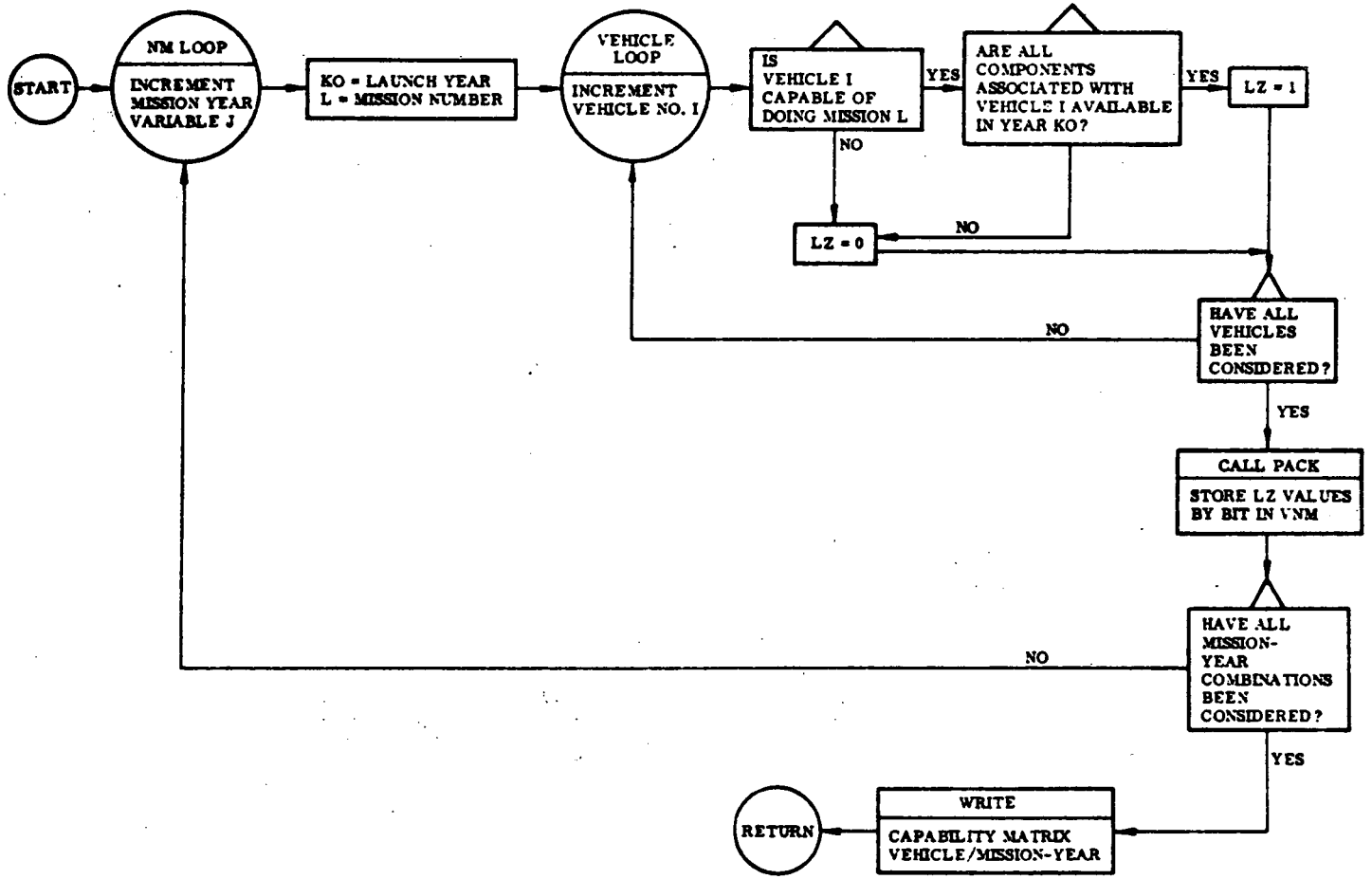
X is the name of the result returned in A4 format

SUBROUTINE ASSIGN



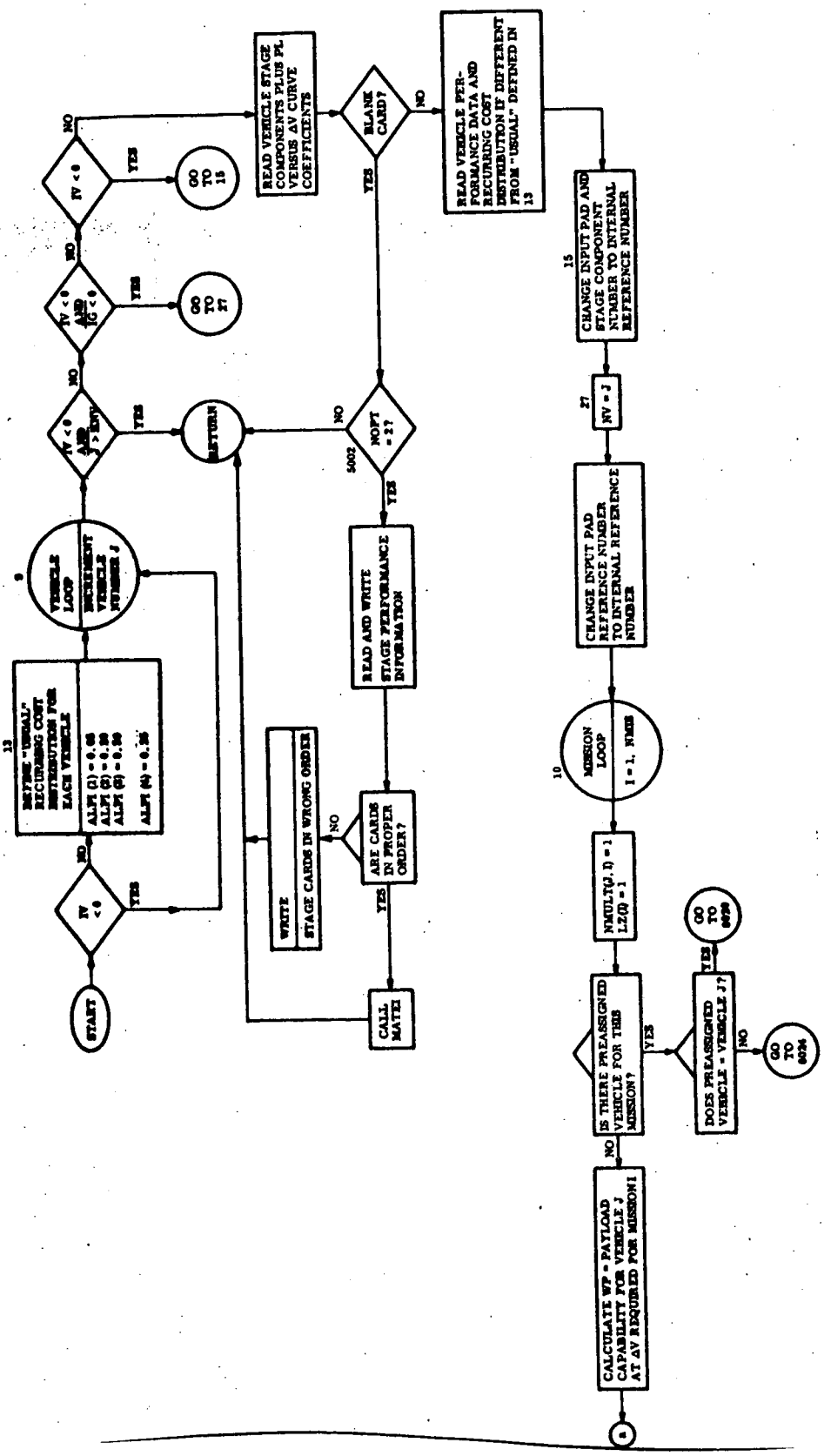
SUBROUTINE ASSIGN (CONT.)

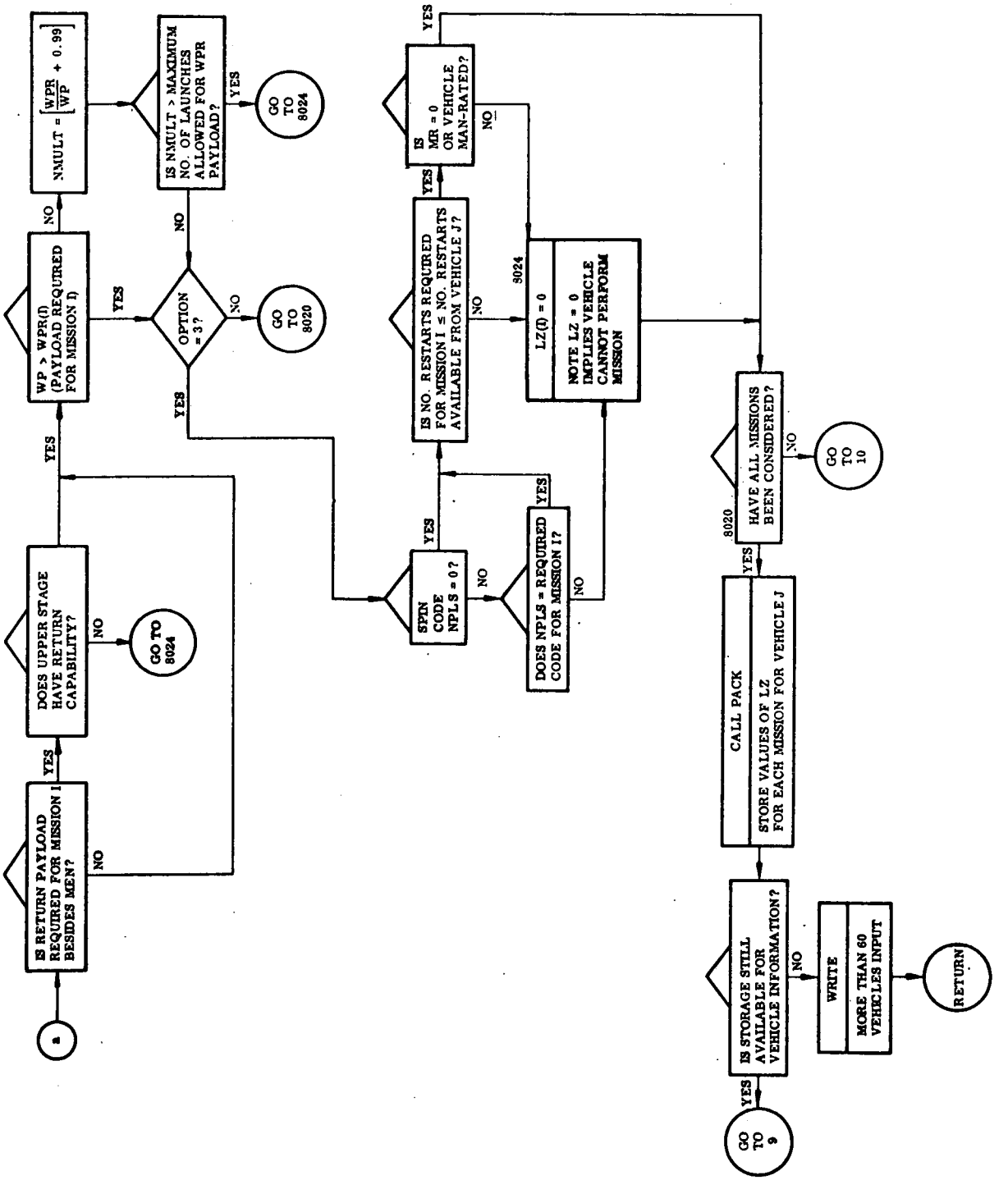




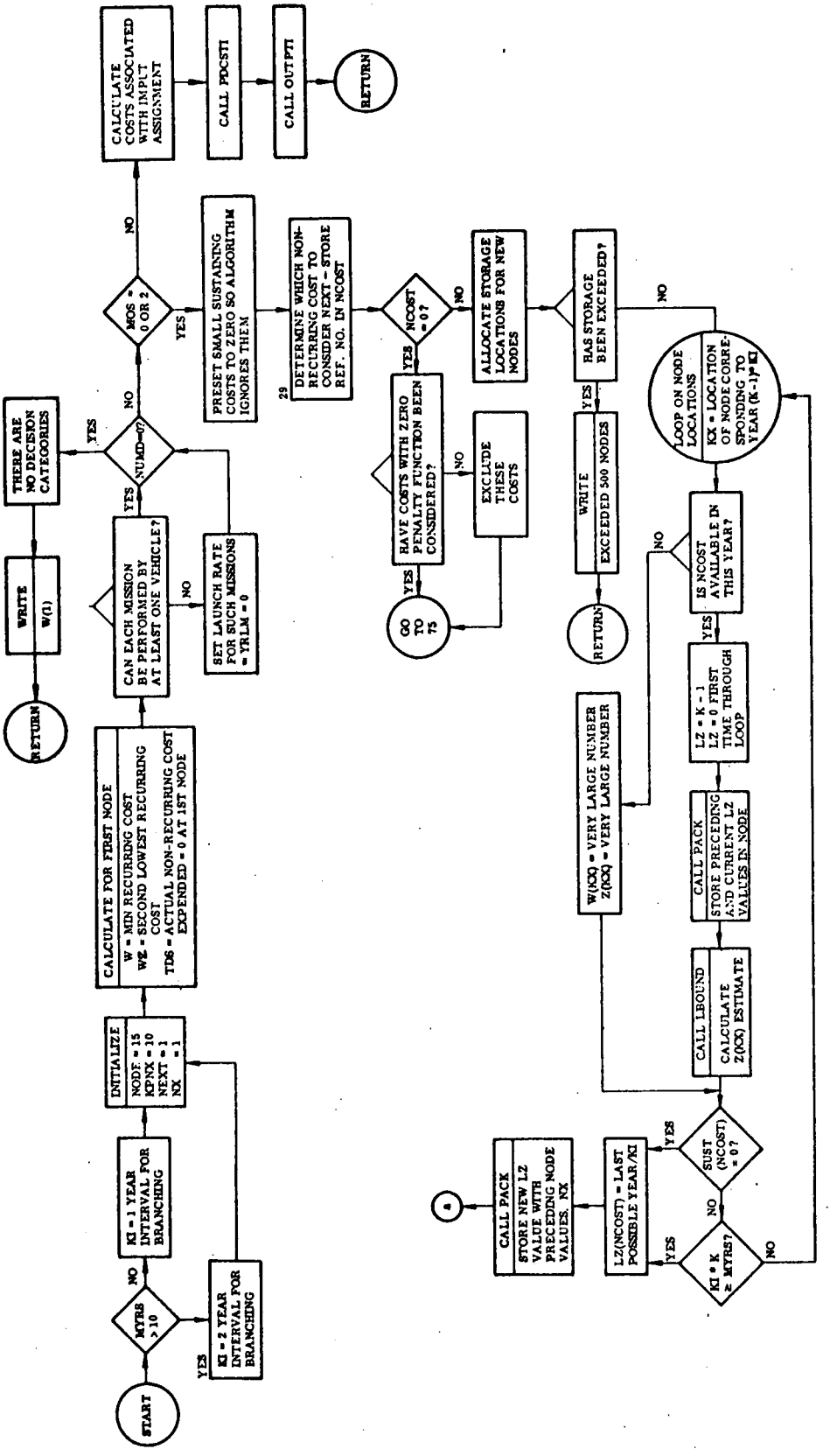
SUBROUTINE AVAILI

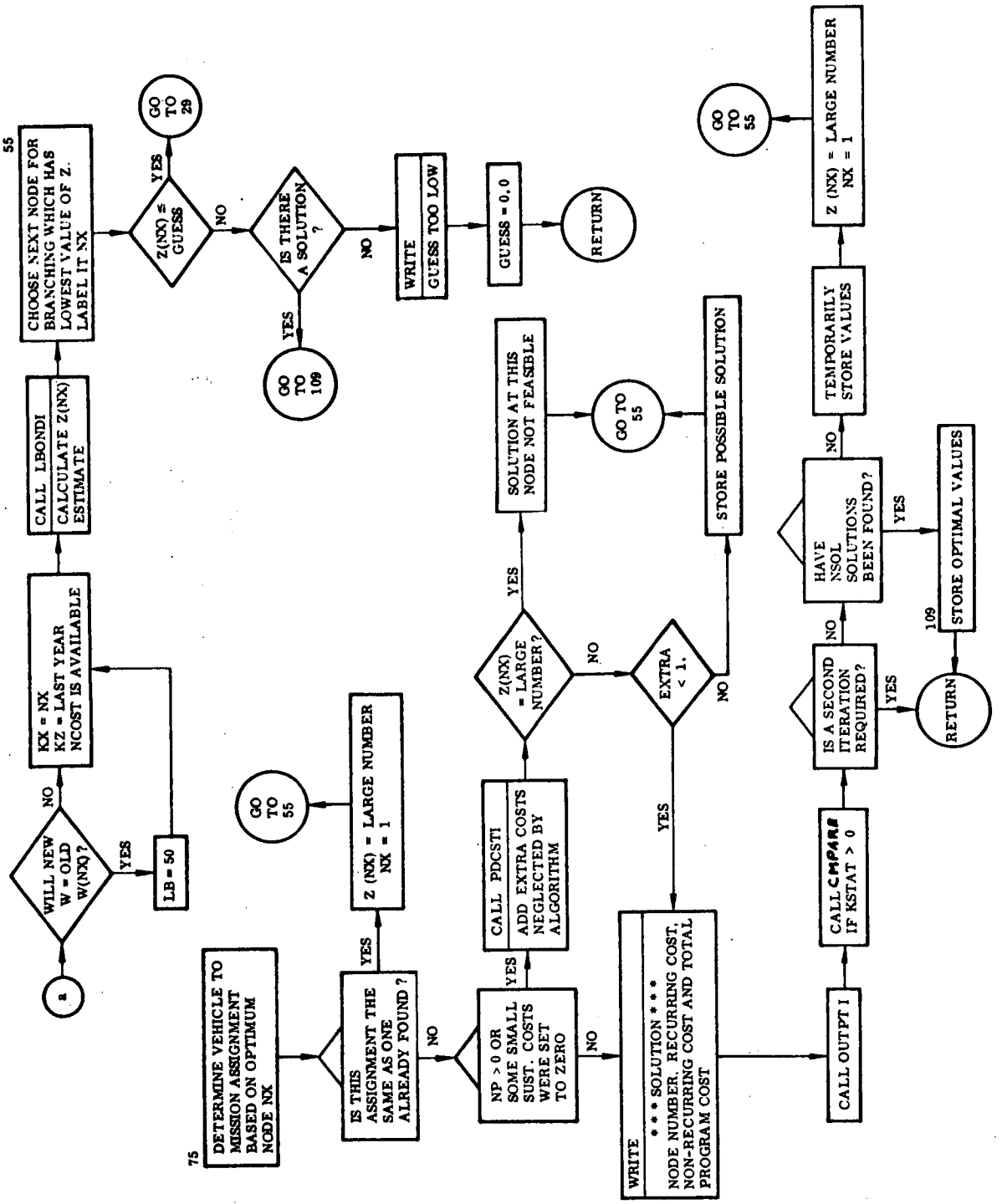
SUBROUTINE CAPBLI



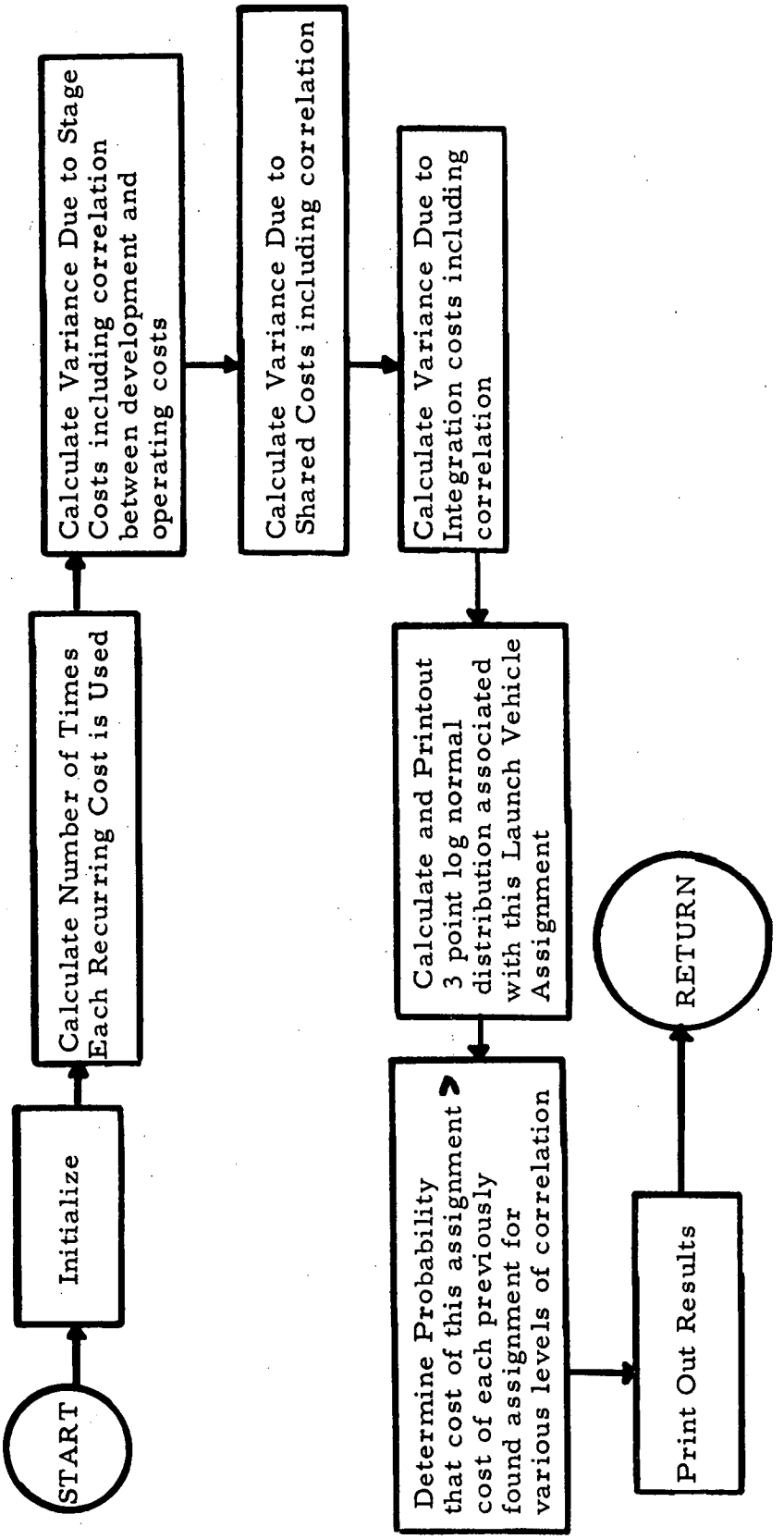

a
IS RETURN PAYLOAD REQUIRED FOR MISSION I BESIDES MEN?
YES
NO
DOES UPPER STAGE HAVE RETURN CAPABILITY?
YES
NO
GO TO 8024
WP > WPR(I) (PAYLOAD REQUIRED FOR MISSION I)
YES
NO
NMULT = [WPR/WP + 0.99]
OPTION = 3?
YES
NO
GO TO 8020
IS NMULT > MAXIMUM NO. OF LAUNCHES ALLOWED FOR WPR PAYLOAD?
YES
NO
GO TO 8024
SPIN CODE NPLS = 0?
YES
NO
DOES NPLS = REQUIRED CODE FOR MISSION I?
YES
NO
IS NO. RESTARTS REQUIRED FOR MISSION I ≤ NO. RESTARTS AVAILABLE FROM VEHICLE J?
YES
NO
IS MR = 0 OR VEHICLE MAN-RATED?
YES
NO
3024
LZ(I) = 0
NOTE LZ = 0 IMPLIES VEHICLE CANNOT PERFORM MISSION
8020
HAVE ALL MISSIONS BEEN CONSIDERED?
YES
NO
GO TO 10
CALL PACK
STORE VALUES OF LZ FOR EACH MISSION FOR VEHICLE J
IS STORAGE STILL AVAILABLE FOR VEHICLE INFORMATION?
YES
NO
WRITE
MORE THAN 60 VEHICLES INPUT
RETURN
GO TO 9

SUBROUTINE CHOOZS

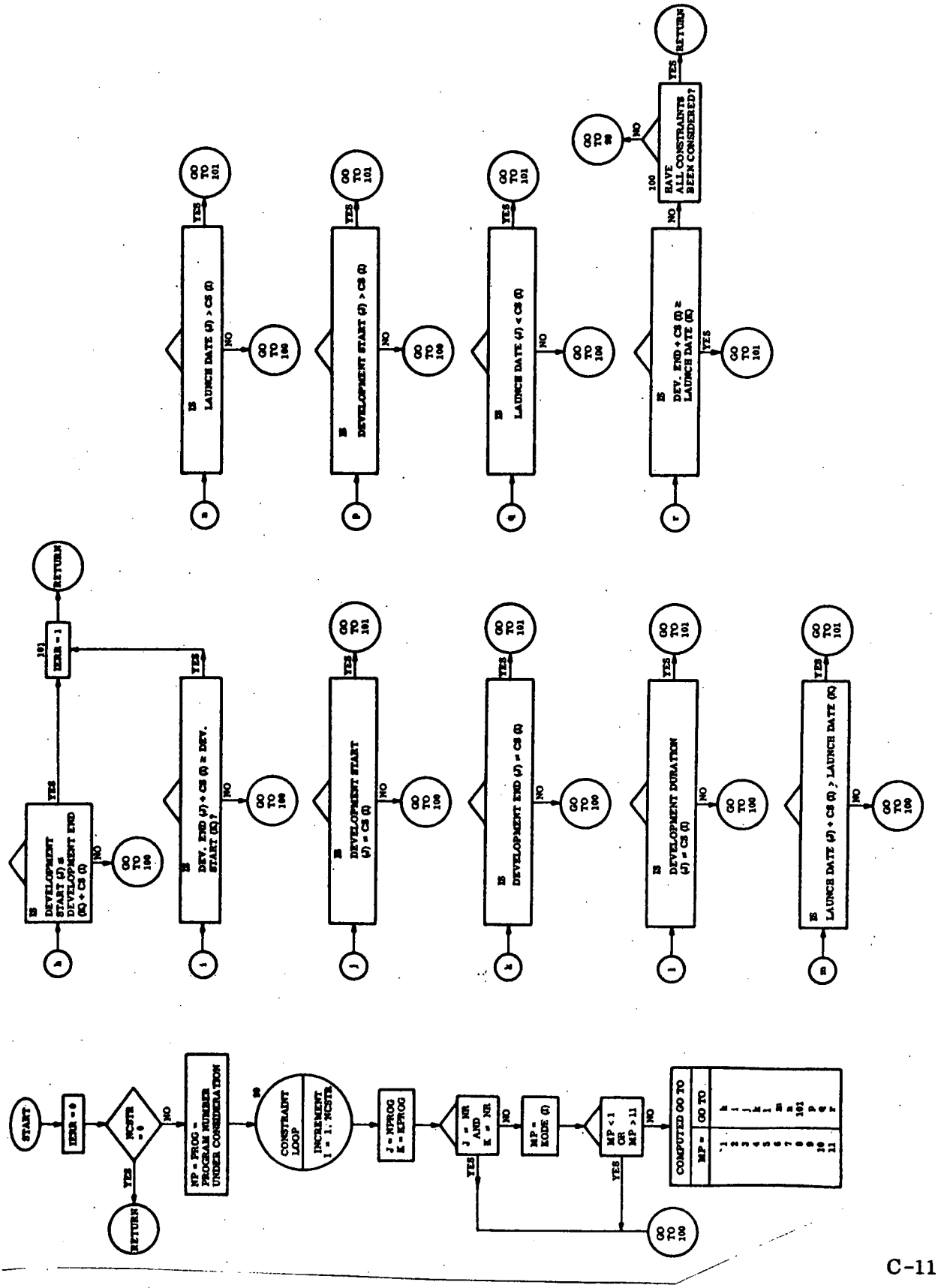


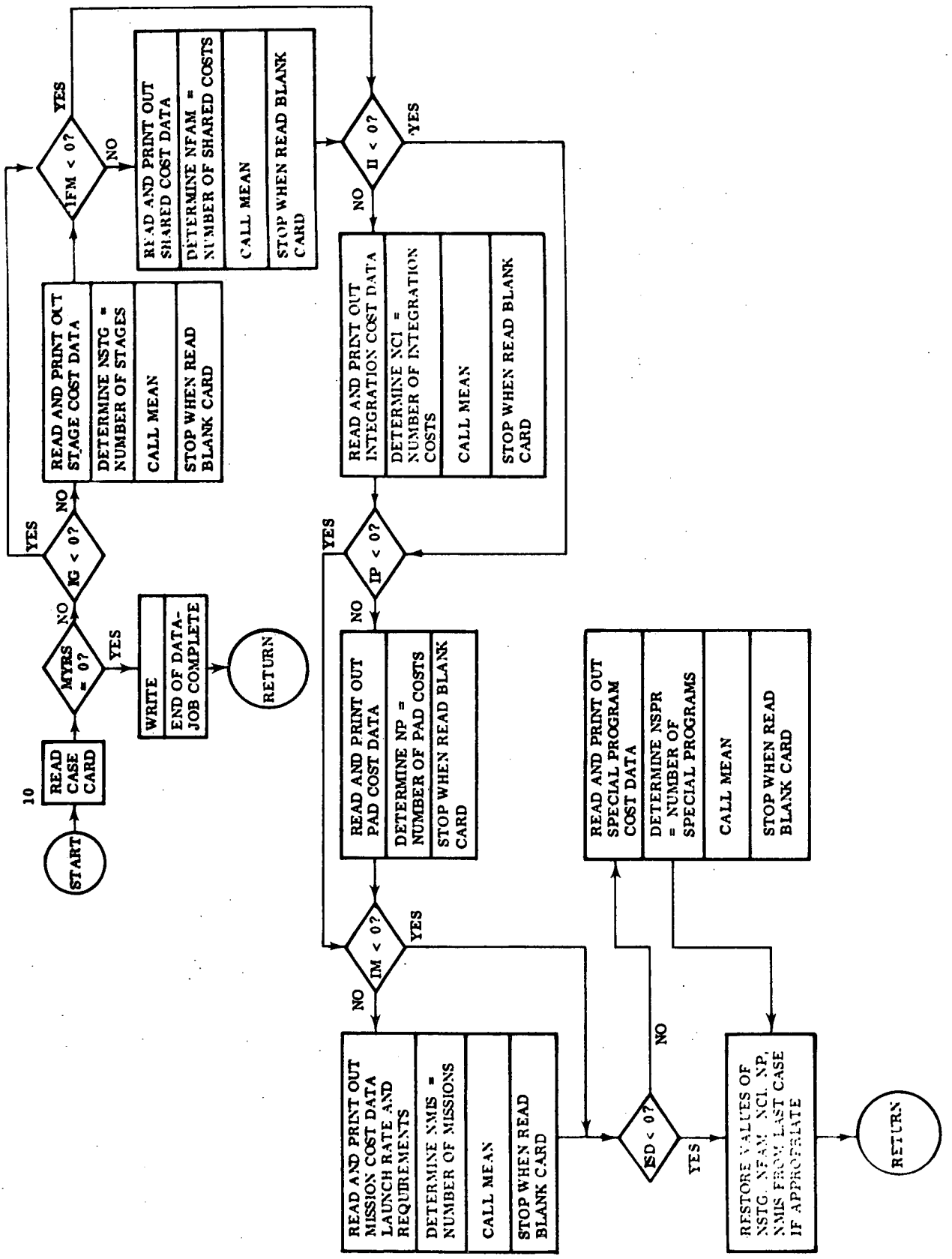

a
WILL NEW W = OLD W(NX)?
YES
LB = 50
NO
KX = NX
KZ = LAST YEAR
NCOST IS AVAILABLE
CALL LBONDI
CALCULATE Z(NX) ESTIMATE
55
CHOOSE NEXT NODE FOR BRANCHING WHICH HAS LOWEST VALUE OF Z. LABEL IT NX
Z(NX) ≤ GUESS
YES
GO TO 29
NO
IS THERE A SOLUTION ?
YES
GO TO 109
NO
WRITE
GUESS TOO LOW
GUESS = 0.0
RETURN
75
DETERMINE VEHICLE TO MISSION ASSIGNMENT BASED ON OPTIMUM NODE NX
IS THIS ASSIGNMENT THE SAME AS ONE ALREADY FOUND ?
YES
GO TO 55
Z (NX) = LARGE NUMBER
NX = 1
NO
NP > 0 OR SOME SMALL SUST. COSTS WERE SET TO ZERO
YES
CALL PDCSTI
ADD EXTRA COSTS NEGLECTED BY ALGORITHM
NO
Z(NX) = LARGE NUMBER ?
YES
SOLUTION AT THIS NODE NOT FEASIBLE
GO TO 55
NO
EXTRA < 1.
YES
WRITE
* * * SOLUTION * * *
NODE NUMBER, RECURRING COST, NON-RECURRING COST AND TOTAL PROGRAM COST
NO
STORE POSSIBLE SOLUTION
CALL OUTPT I
CALL CMPARE IF KSTAT > 0
IS A SECOND ITERATION REQUIRED?
NO
HAVE NSOL SOLUTIONS BEEN FOUND?
NO
TEMPORARILY STORE VALUES
Z (NX) = LARGE NUMBER
NX = 1
GO TO 55
YES
109
STORE OPTIMAL VALUES
RETURN
YES

SUBROUTINE CMPARE

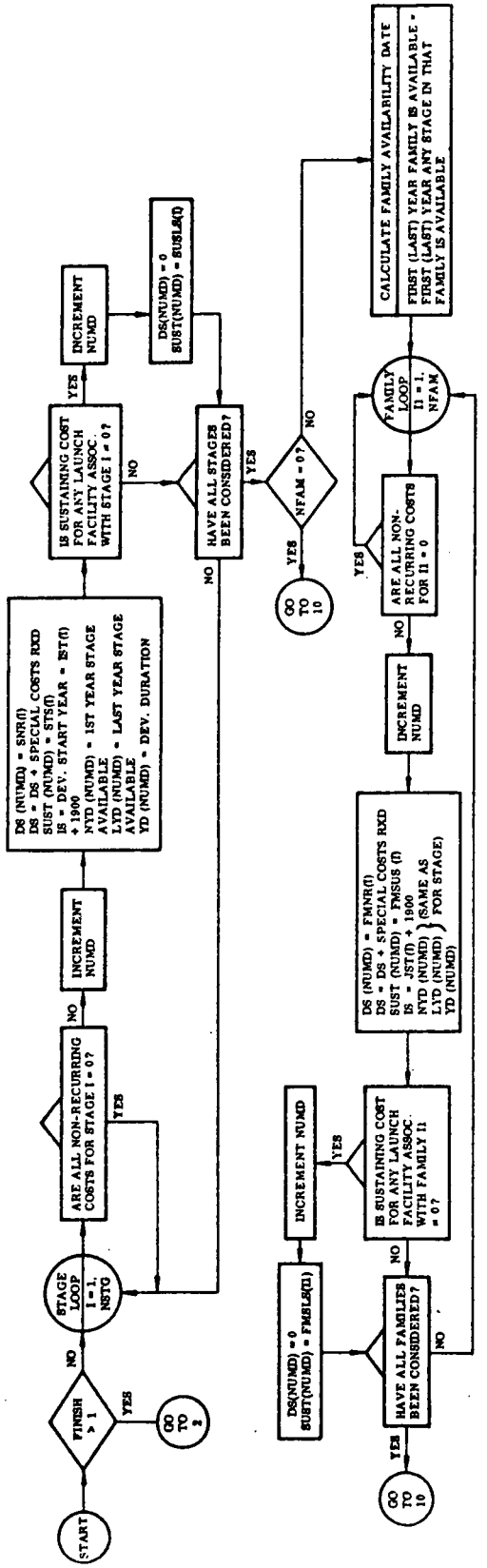


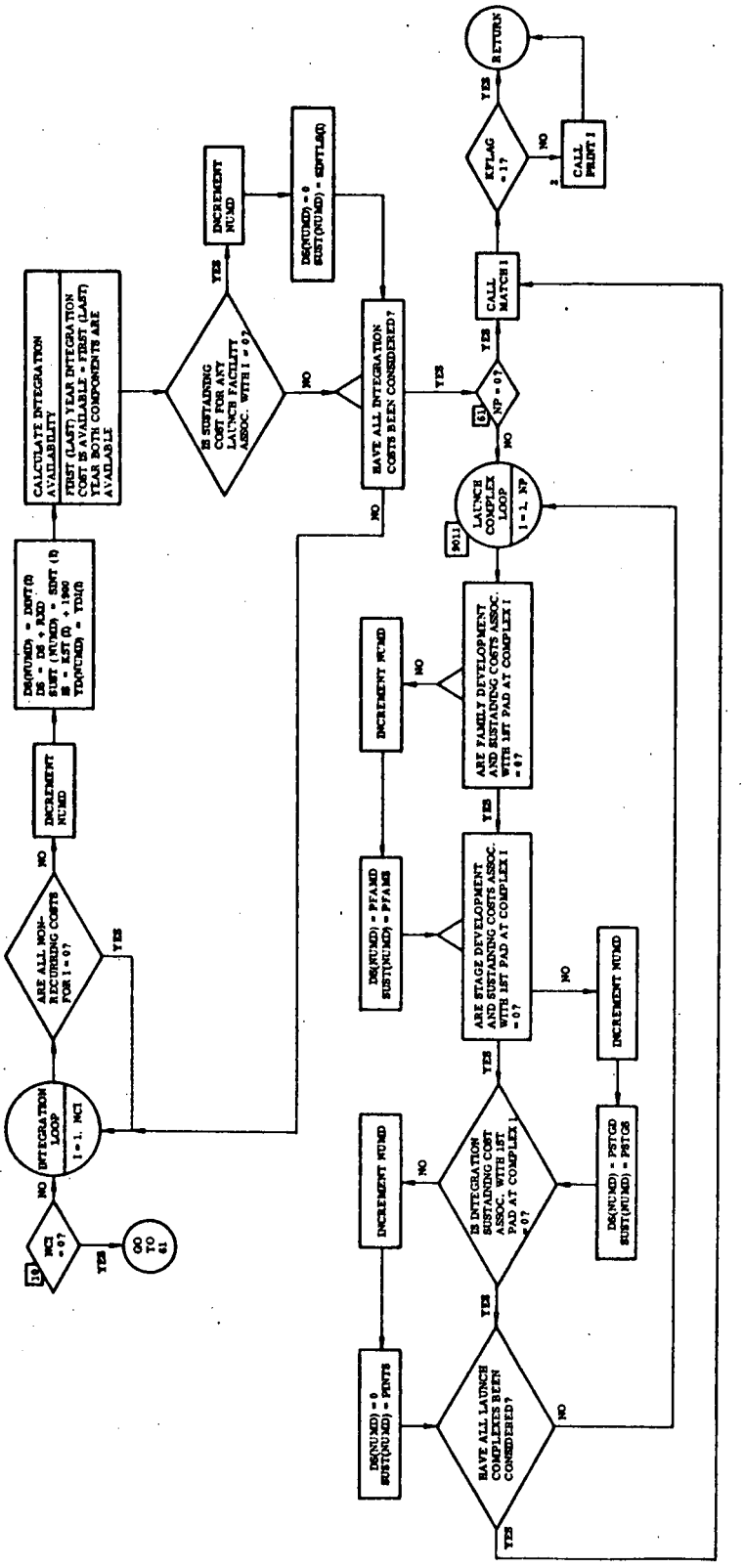
SUBROUTINE CONSTR




START
10
READ CASE CARD
MYRS = 0?
YES
WRITE
END OF DATA-JOB COMPLETE
RETURN
NO
MG < 0?
YES
NO
READ AND PRINT OUT STAGE COST DATA
DETERMINE NSTG = NUMBER OF STAGES
CALL MEAN
STOP WHEN READ BLANK CARD
IFM < 0?
YES
NO
READ AND PRINT OUT SHARED COST DATA
DETERMINE NFAM = NUMBER OF SHARED COSTS
CALL MEAN
STOP WHEN READ BLANK CARD
II < 0?
YES
NO
READ AND PRINT OUT INTEGRATION COST DATA
DETERMINE NCI = NUMBER OF INTEGRATION COSTS
CALL MEAN
STOP WHEN READ BLANK CARD
IP < 0?
YES
NO
READ AND PRINT OUT PAD COST DATA
DETERMINE NP = NUMBER OF PAD COSTS
STOP WHEN READ BLANK CARD
IM < 0?
YES
NO
READ AND PRINT OUT MISSION COST DATA LAUNCH RATE AND REQUIREMENTS
DETERMINE NMIS = NUMBER OF MISSIONS
CALL MEAN
STOP WHEN READ BLANK CARD
ISD < 0?
YES
NO
READ AND PRINT OUT SPECIAL PROGRAM COST DATA
DETERMINE NSPR = NUMBER OF SPECIAL PROGRAMS
CALL MEAN
STOP WHEN READ BLANK CARD
RESTORE VALUES OF NSTG, NFAM, NCI, NP, NMIS FROM LAST CASE IF APPROPRIATE
RETURN

SUBROUTINE DECSNI




RETURN
YES
KFLAG = 1?
NO
CALL PRINT 1
CALL MATCH 1
YES
NP = 0?
NO
LAUNCH COMPLEX LOOP
I = 1, NP
ARE FAMILY DEVELOPMENT AND SUSTAINING COSTS ASSOC. WITH 1ST PAD AT COMPLEX I = 0?
NO
INCREMENT NUMD
YES
ARE STAGE DEVELOPMENT AND SUSTAINING COSTS ASSOC. WITH 1ST PAD AT COMPLEX I = 0?
NO
INCREMENT NUMD
YES
IS INTEGRATION SUSTAINING COST ASSOC. WITH 1ST PAD AT COMPLEX I = 0?
NO
INCREMENT NUMD
YES
HAVE ALL LAUNCH COMPLEXES BEEN CONSIDERED?
NO
YES
HAVE ALL INTEGRATION COSTS BEEN CONSIDERED?
NO
YES
IS SUSTAINING COST FOR ANY LAUNCH FACILITY ASSOC. WITH I = 0?
NO
YES
INCREMENT NUMD
CALCULATE INTEGRATION AVAILABILITY
FIRST (LAST) YEAR INTEGRATION COST IS AVAILABLE = FIRST (LAST) YEAR BOTH COMPONENTS ARE AVAILABLE
ARE ALL NON-RECURRING COSTS FOR I = 0?
NO
YES
INCREMENT NUMD
INTEGRATION LOOP
I = 1, NCI
NCI = 0?
YES
GO TO 81

SUBROUTINE INPUT

IDENTIFICATION

AL INPT Generalized Data Input Subroutine
360/Assembler Language
Written by R. E. Slye

PURPOSE

This subroutine provides for input of single-precision fixed and floating point numbers and Hollerith information. Usage is particularly convenient inasmuch as no format statements are required, and data may be loaded in any order irrespective of the order in the calling statement.

USAGE

The calling statement is

CALL INPUT (5HALPHA, ALPHA, 4BETA, BETA, ...)

In the above, the Hollerith literals represent the external names of variables or arrays as they should appear on data cards. The other arguments are the internal names of the variables and arrays as referenced in the source program. It will become apparent that by using the external names in addition to the symbolic location names, it is possible to enter data for a variable on an input card without regard to its relative location in the calling sequence of the program.

ACCEPTABLE INPUT DATA FORMS

A. Floating Point General Form

	Examples
Up to 9 decimal digits, with a decimal point permitted at the beginning, at the end or between two digits. A preceding plus or minus sign is optional. A decimal exponent preceded by E+ or + or - if negative may follow. If no decimal point appears, the exponent is mandatory. The magnitude of the number must be between the approximate limits of 10^{-75} and 10^{75}.	17. 5.0 -.0003 5.0E3 (5.0×10^{3}) 5.0E+3 (5.0×10^{3}) 5.0E-7 (5.0×10^{-7})

B. Decimal Integers General Form

	Examples
The magnitude of the number must be less than 2^{31}. A preceding plus or minus sign is optional.	3 +1 -28987

C. Hollerith Information General Form

	Examples
Any number of characters, including blanks. The number of characters is specified by writing nH preceding the Hollerith information. n is the number of characters in the block following nH.	14HTHIS IS A TEST 6HALPHA

RULES FOR PREPARATION OF DATA CARDS

Blanks are ignored except within Hollerith data fields.

Data must be contained within card columns 1 through 72.

It is not necessary that variable names on the data cards appear in the same order as those in the calling sequence. The routine will search the list for the name and its core location.

Individual data items are separated by commas.

An equal sign separates the name of a variable and its first data item.

A comma separates the end of a data set and the next variable name.

A data input record is terminated by an asterisk (*).

It is not necessary to input a data set for each name in the calling sequence.

Elements of an array may be skipped by writing consecutive commas — i.e., no data between the commas; or by singly subscripting the array name. Double subscripting is illegal. Thus, if it is desired to input data into a three-element vector V, one could write:

```
V = 2.79,,1.32
```

No data would be entered into V(2). What was originally there remains there. Alternatively, the above could be written:

```
V(1) = 2.79, V(3) = 1.32
```

Special Feature. The card image is normally written on the system output unit, tape 6, prior to being processed by the routine. If an N is punched in column 73, the card will not be listed. If column 73 contains a C, the card is treated as a comment only; i.e., it is not scanned for data. If the card contains CE in columns 73—74, the card will be treated as a comment card, and a page will be ejected.

EXAMPLE

If the following call statement appeared in a FORTRAN program,

CALL INPUT (1HA, A, 1HB, B, 1HC, C, 1HD, D, 1HP, P, 1HR, R, 1HS, S)
the input cards could be punched as follows:

```
A    = 3.14159265, B = 707, C = 1870,      1st card
D    = 1., 2., 3., 4., 5., 6., 7., 8., 9., 2nd card
R(2) = 3, R(5) = 74., 42,                  3rd card
F    = 22HTHIS IS A CHECKOUT RUN*          4th card
```

Note that D must be dimensioned at least 9, R dimensioned at least 7 and P at least 6.

Also R(1), R(3), R(4), and R(6) are unchanged.

Even though S appears in the CALL statement, it is not necessary that it appear on one of the input cards. The * on card 4 signifies the end of the data record. This means that the routine will return control to the calling program.

RESTRICTIONS

The following errors will be detected by the subroutine. A diagnostic message and the card in error will be permited on the system output unit, tape 6.

1. Name on data card exceeds six characters.
2. Name on data card does not appear in the calling sequence.
3. Punctuation errors.
4. Name on data card begins with a non-alphabetic character.
5. Decimal or integer data out of range.

This subroutine may be used for reading double precision numbers; however, only the high order part of the number will be loaded. To clear the low order part of the number, write

```
DWORD = 1., 0,
```

ADDITIONAL INFORMATION

1. A slash (/) on a data card (not in an H field) indicates that information to the right of the slash is not to be scanned for data. Therefore, these columns may be used for comments.
2. In addition to the above means for entering Hollerith information, Hollerith may also be entered by enclosing it in apostrophes, i.e., P = 'THIS IS A CHECKOUT RUN'
3. Floating point and integer data may be repeated into consecutive locations by use of the letter X followed by the data; i.e.,

   D = 1., 4X2., 3.,

   is equivalent to

   D = 1., 2., 2., 2., 2., 3.,

4. Alphanumeric data may also be repeated. The use of the letter X is optional. For example, to set an array dimensioned 18 to blanks, write

   TITLE = 18' ',

   If the alphanumeric field exceeds 4 characters, only the last word will be repeated. For example,

   DATA = 3'ABCDEF', will result in
   ABCDEF EF EF

5. If a name on a data card is not followed by an equal sign, it will be retrieved from the calling program. For example, if in the calling program, X and ALPHA are dimensioned at least 2, then the following data card

   X = 3.1, ALPHA(2),

   will result in the current value of ALPHA(2) being stored in X(2).

As an additional example, suppose that the calling FORTRAN program has the following sequence:

```
LOGICAL
 . . .
 . . .
TRUE = .TRUE.
FALSE = .FALSE.
 . . .
 . . .
CALL INPUT (. . . , 'OK', OK, 'TRUE', TRUE,
                    'FALSE', FALSE, . . .)
```

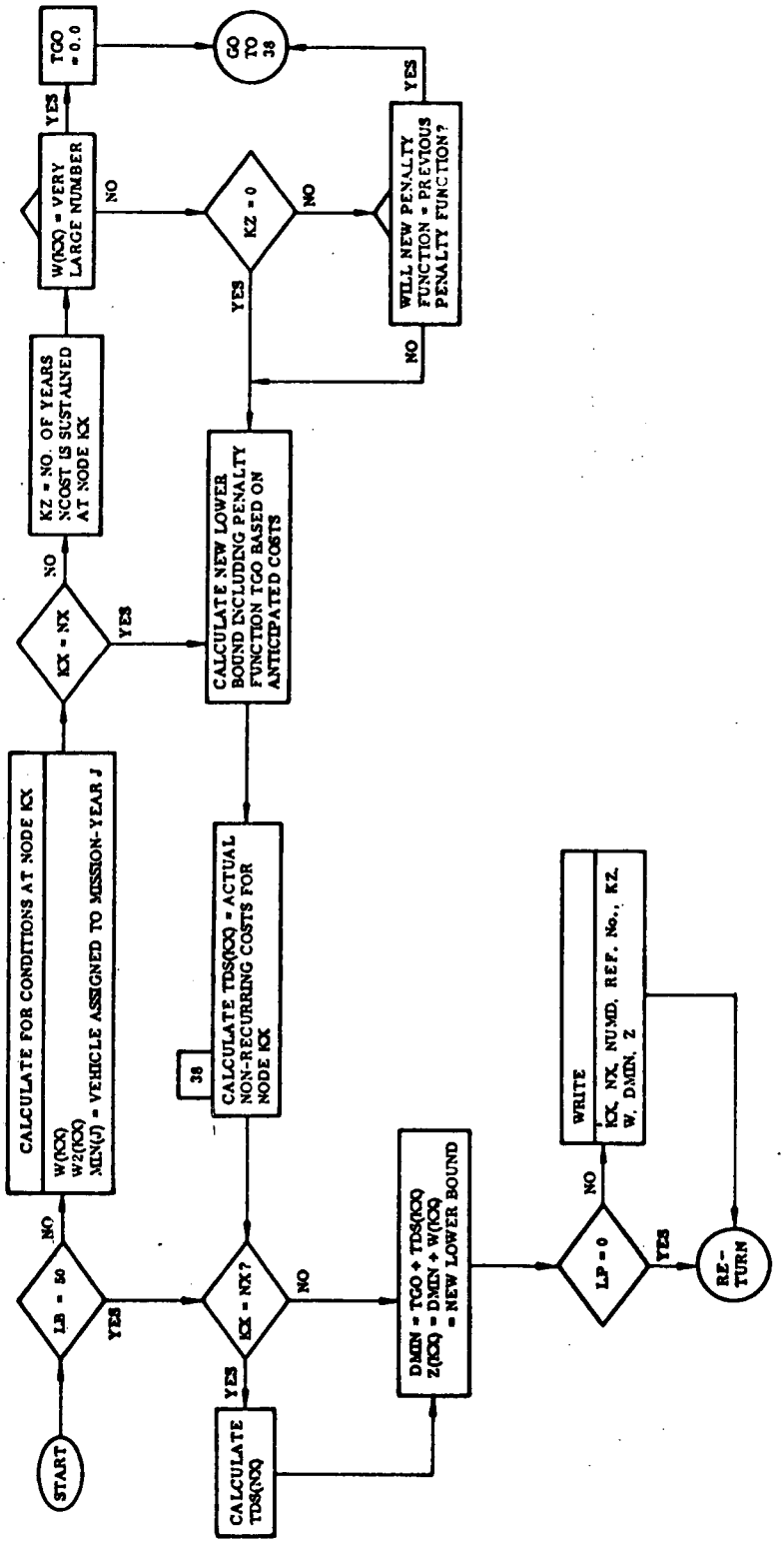
Then a data card written as follows,

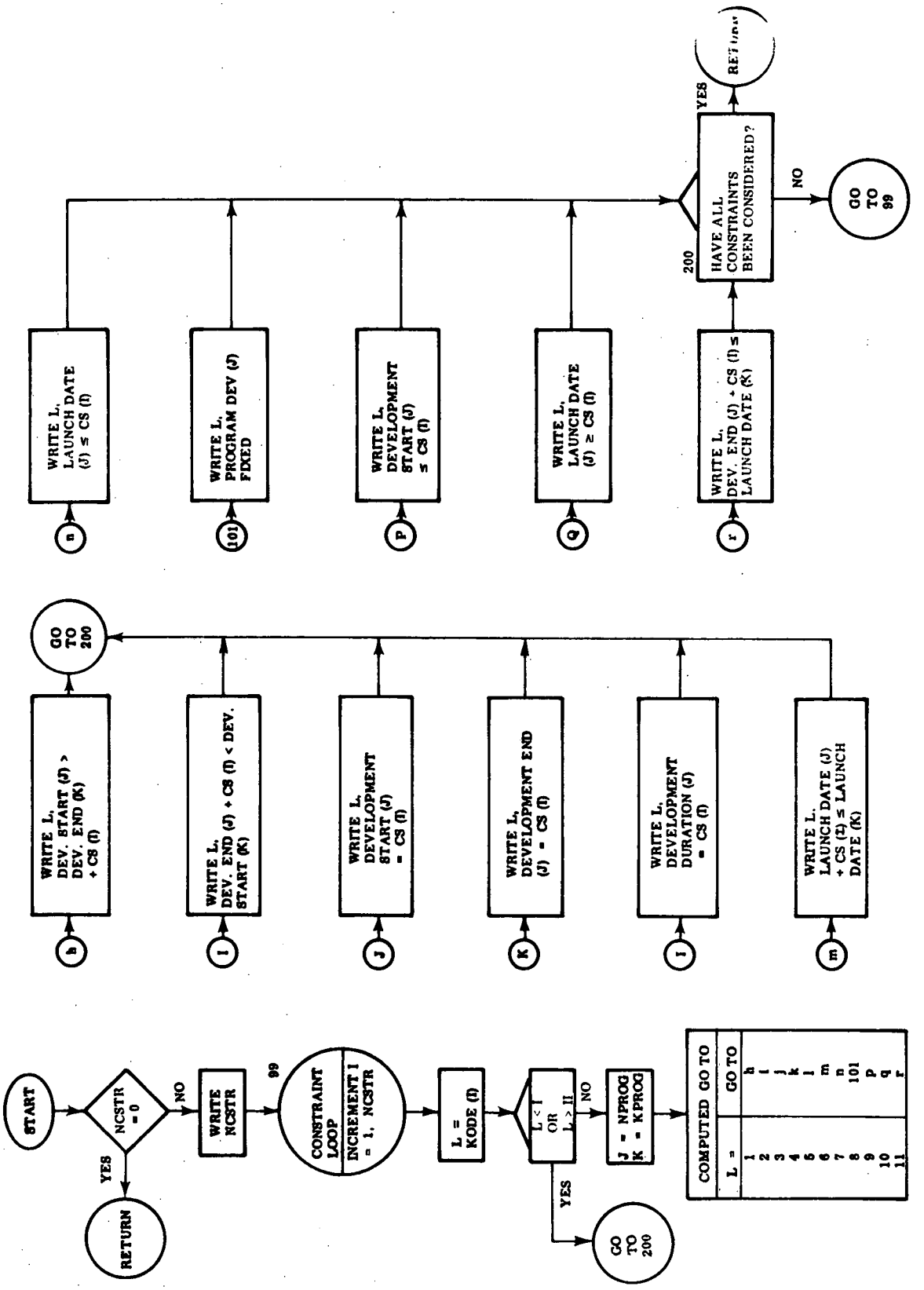
```
OK = TRUE,
```

will result in the input of logical data to the program.

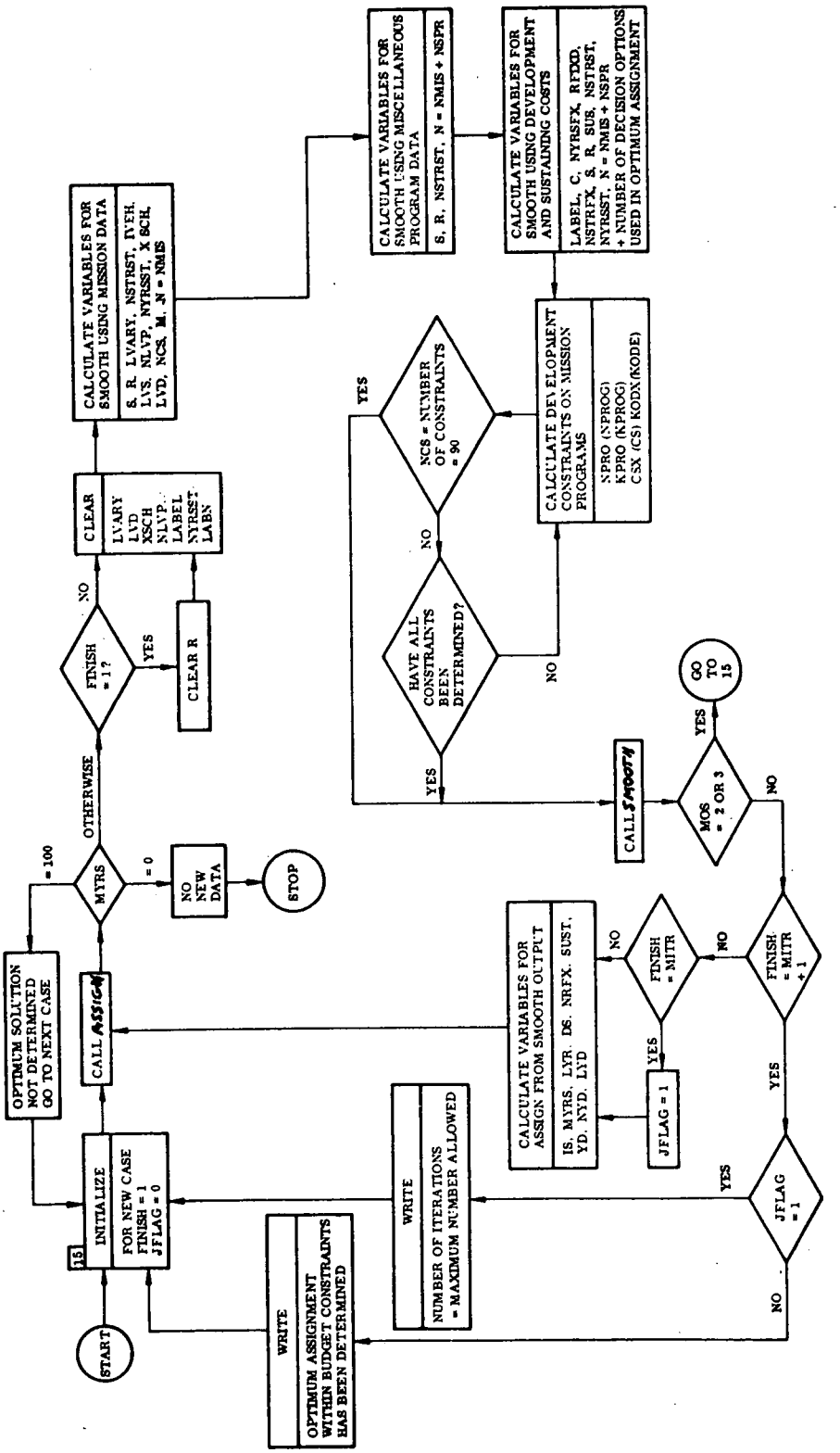
6. If a comma is omitted from a data card, a warning will be written on the system output unit and execution will continue. However, for any other type of error, execution will be suppressed, and the remaining data cards will be scanned for errors.
7. This subroutine will accept data cards punched on either a 026 for 029 keypunch.

SUBROUTINE LBONDI

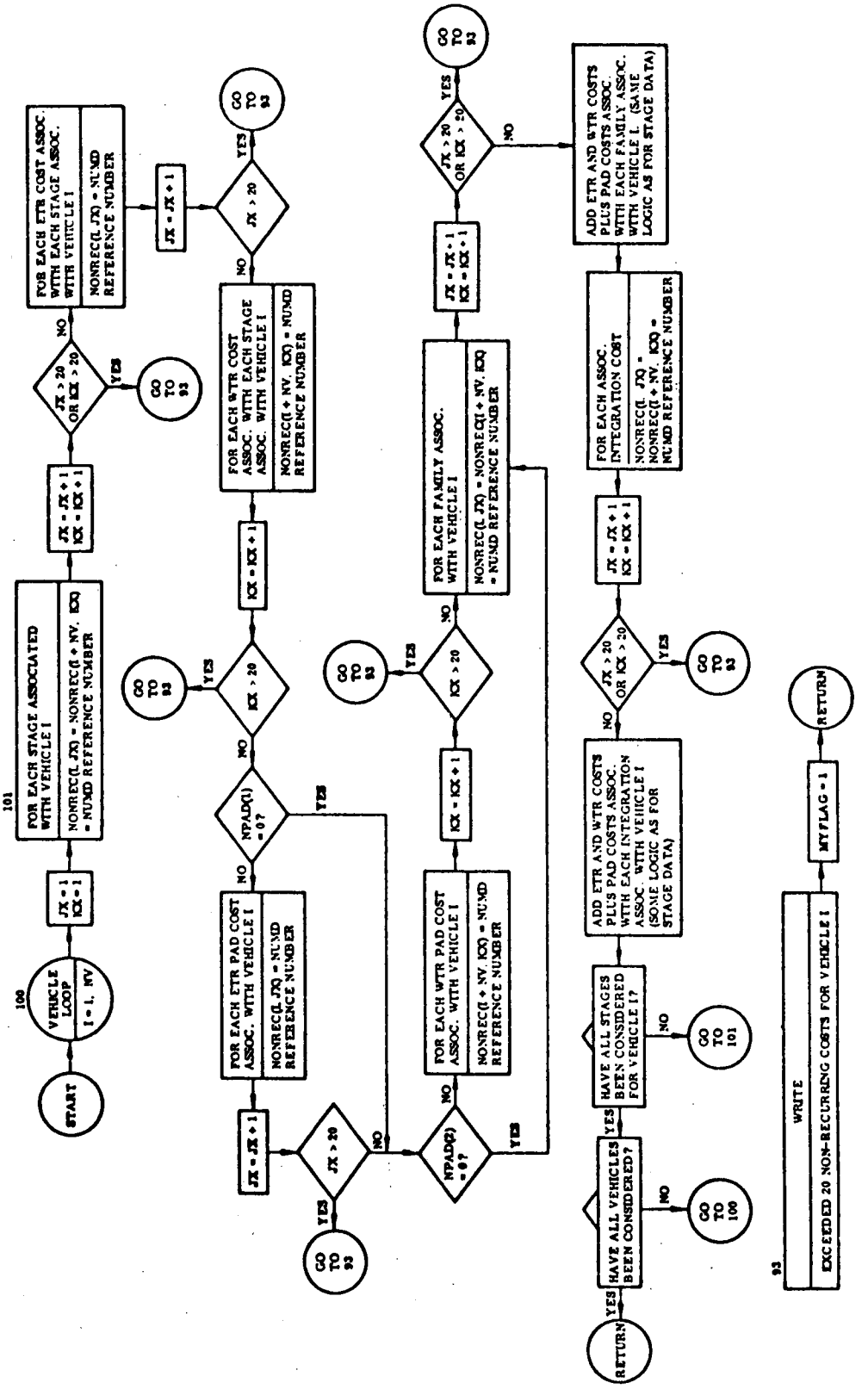


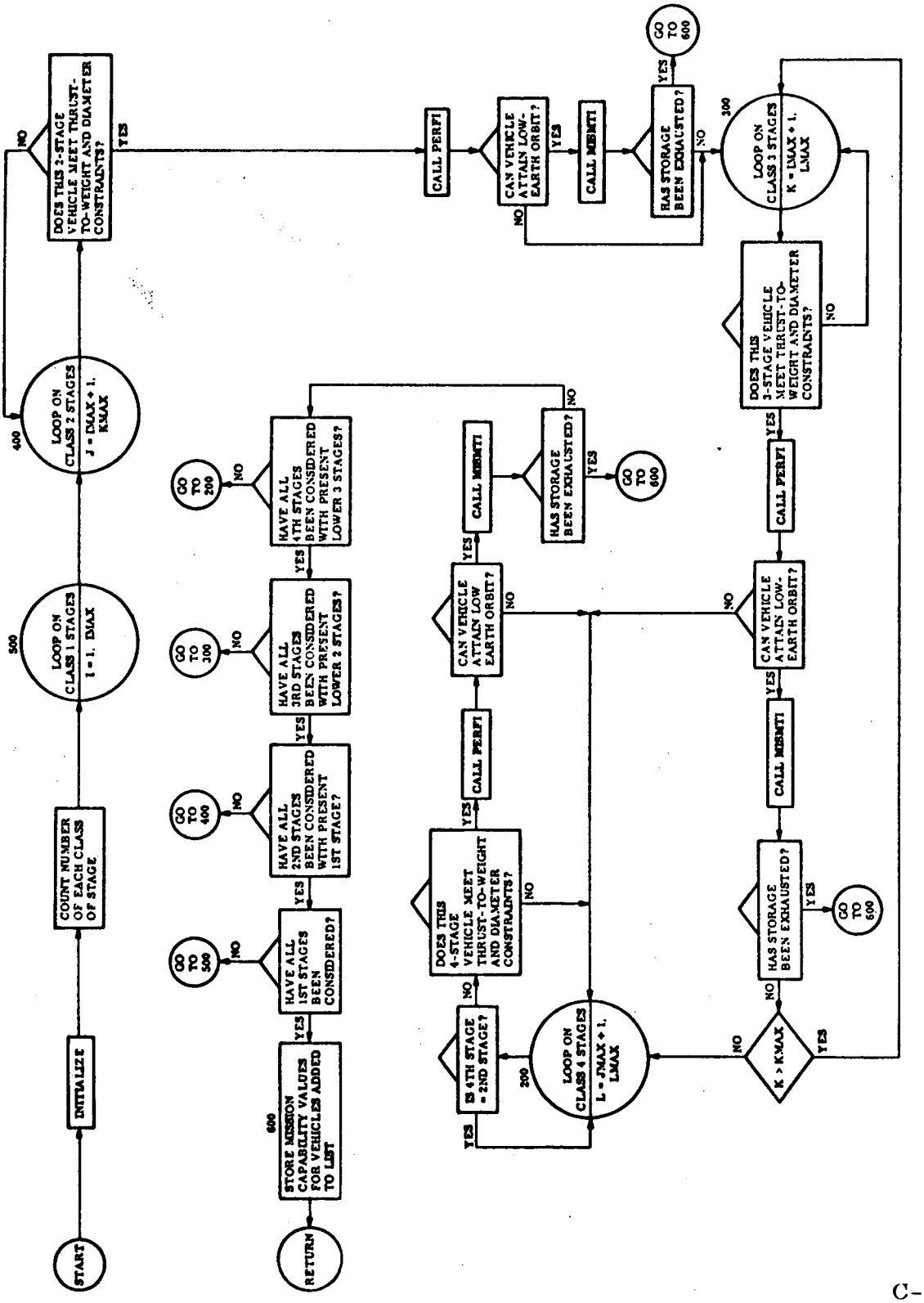

START
NCSTR = 0
YES
RETURN
NO
WRITE NCSTR
99
CONSTRAINT LOOP
INCREMENT I = 1, NCSTR
L = KODE (I)
L < 1 OR L ≥ 11
YES
GO TO 200
NO
J = NPROG
K = KPROG
COMPUTED GO TO
L = | GO TO
1 | h
2 | i
3 | j
4 | k
5 | l
6 | m
7 | n
8 | 101
9 | p
10 | q
11 | r
WRITE L, DEV. START (J) > DEV. END (K) + CS (I)
WRITE L, DEV. END (J) + CS (I) < DEV. START (K)
WRITE L, DEVELOPMENT START (J) = CS (I)
WRITE L, DEVELOPMENT END (J) = CS (I)
WRITE L, DEVELOPMENT DURATION (J) = CS (I)
WRITE L, LAUNCH DATE (J) + CS (I) ≤ LAUNCH DATE (K)
GO TO 200
WRITE L, LAUNCH DATE (J) ≤ CS (I)
WRITE L, PROGRAM DEV (J) FIXED
WRITE L, DEVELOPMENT START (J) ≤ CS (I)
WRITE L, LAUNCH DATE (J) ≥ CS (I)
WRITE L, DEV. END (J) + CS (I) ≤ LAUNCH DATE (K)
200
HAVE ALL CONSTRAINTS BEEN CONSIDERED?
YES
NO
GO TO 99
a
101
P
Q
r
b
l
J
K
l
m

SUBROUTINE MASTER

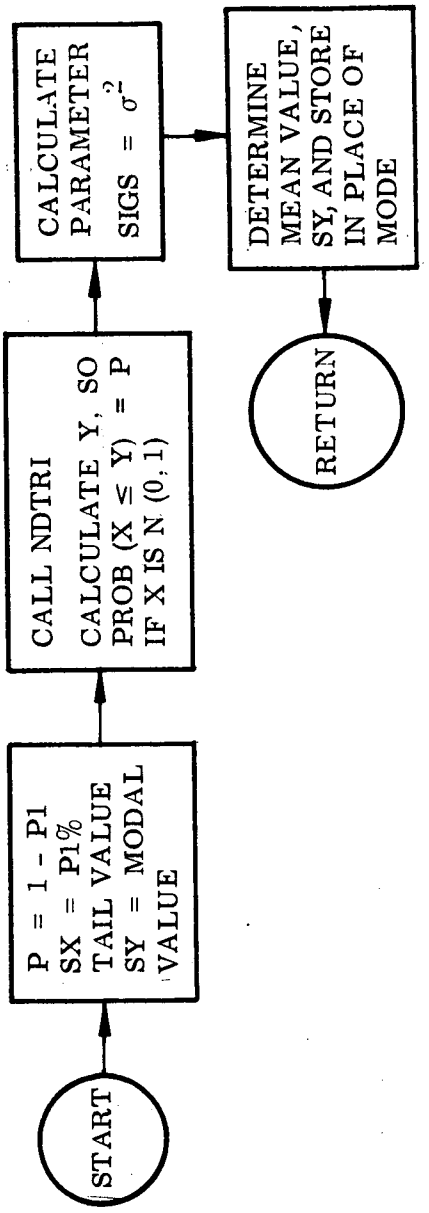


SUBROUTINE MATCHI

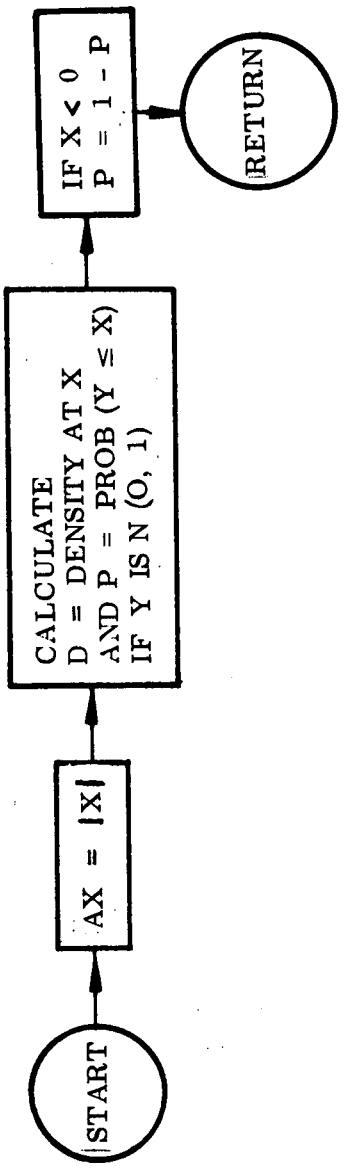



START
INITIALIZE
COUNT NUMBER OF EACH CLASS OF STAGE
500
LOOP ON CLASS 1 STAGES I = 1, IMAX
400
LOOP ON CLASS 2 STAGES J = JMAX + 1, KMAX
DOES THIS 2-STAGE VEHICLE MEET THRUST-TO-WEIGHT AND DIAMETER CONSTRAINTS?
NO
YES
CALL PERFI
CAN VEHICLE ATTAIN LOW-EARTH ORBIT?
NO
YES
CALL MEMTI
HAS STORAGE BEEN EXHAUSTED?
YES
GO TO 600
NO
300
LOOP ON CLASS 3 STAGES K = LMAX + 1, LMAX
DOES THIS 3-STAGE VEHICLE MEET THRUST-TO-WEIGHT AND DIAMETER CONSTRAINTS?
NO
YES
CALL PERFI
CAN VEHICLE ATTAIN LOW-EARTH ORBIT?
NO
YES
CALL MEMTI
HAS STORAGE BEEN EXHAUSTED?
YES
GO TO 600
NO
K > KMAX
NO
YES
200
LOOP ON CLASS 4 STAGES L = JMAX + 1, LMAX
IS 4TH STAGE = 2ND STAGE?
YES
NO
DOES THIS 4-STAGE VEHICLE MEET THRUST-TO-WEIGHT AND DIAMETER CONSTRAINTS?
YES
NO
CALL PERFI
CAN VEHICLE ATTAIN LOW EARTH ORBIT?
NO
YES
CALL MEMTI
HAS STORAGE BEEN EXHAUSTED?
YES
GO TO 600
NO
HAVE ALL 4TH STAGES BEEN CONSIDERED WITH PRESENT LOWER 3 STAGES?
NO
GO TO 200
YES
HAVE ALL 3RD STAGES BEEN CONSIDERED WITH PRESENT LOWER 2 STAGES?
NO
GO TO 300
YES
HAVE ALL 2ND STAGES BEEN CONSIDERED WITH PRESENT 1ST STAGE?
NO
GO TO 400
YES
HAVE ALL 1ST STAGES BEEN CONSIDERED?
NO
GO TO 500
YES
600
STORE MISSION CAPABILITY VALUES FOR VEHICLES ADDED TO LIST
RETURN

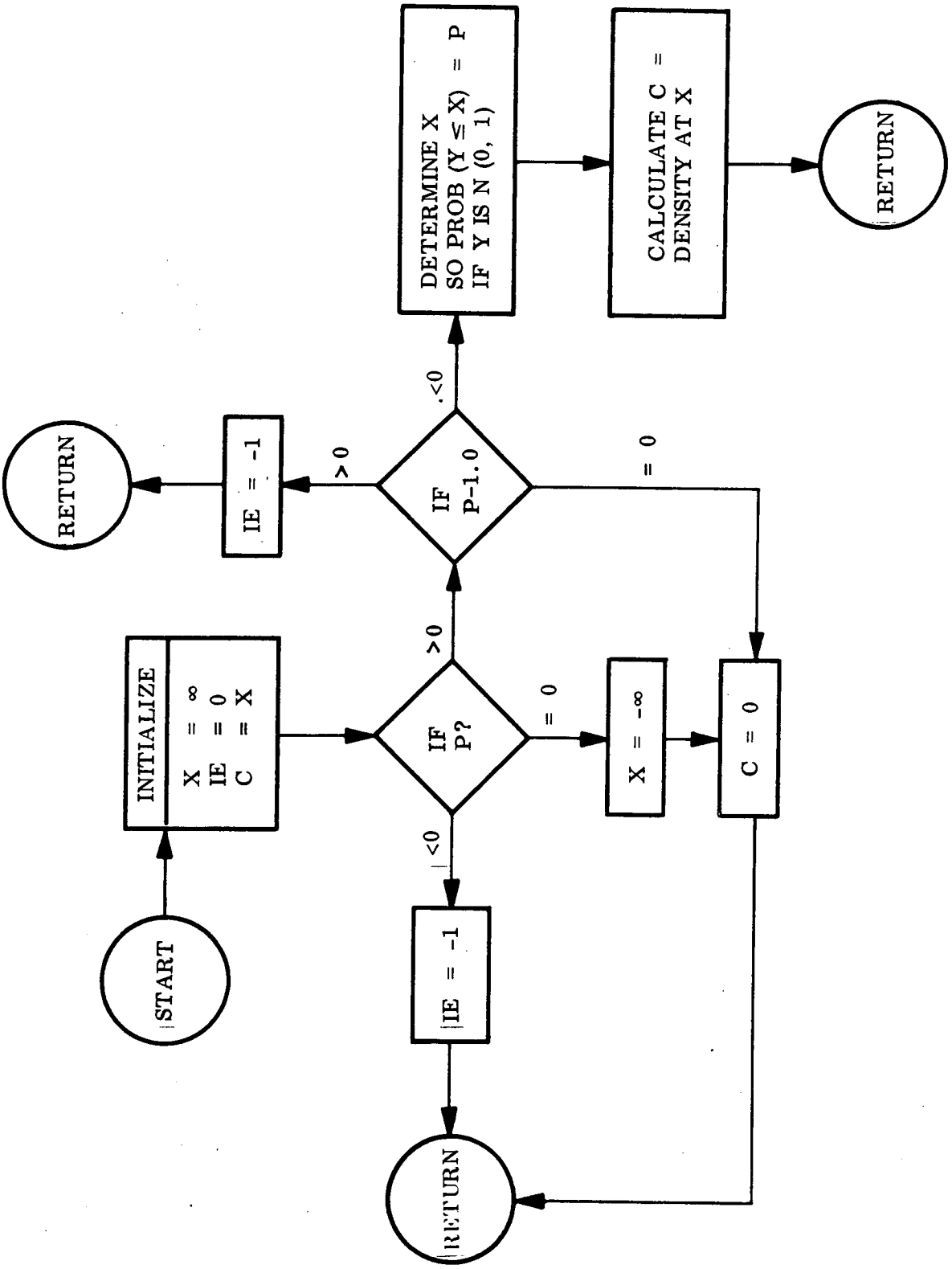
SUBROUTINE MEAN (P1, KSTAT, SIGS, SX, SY)



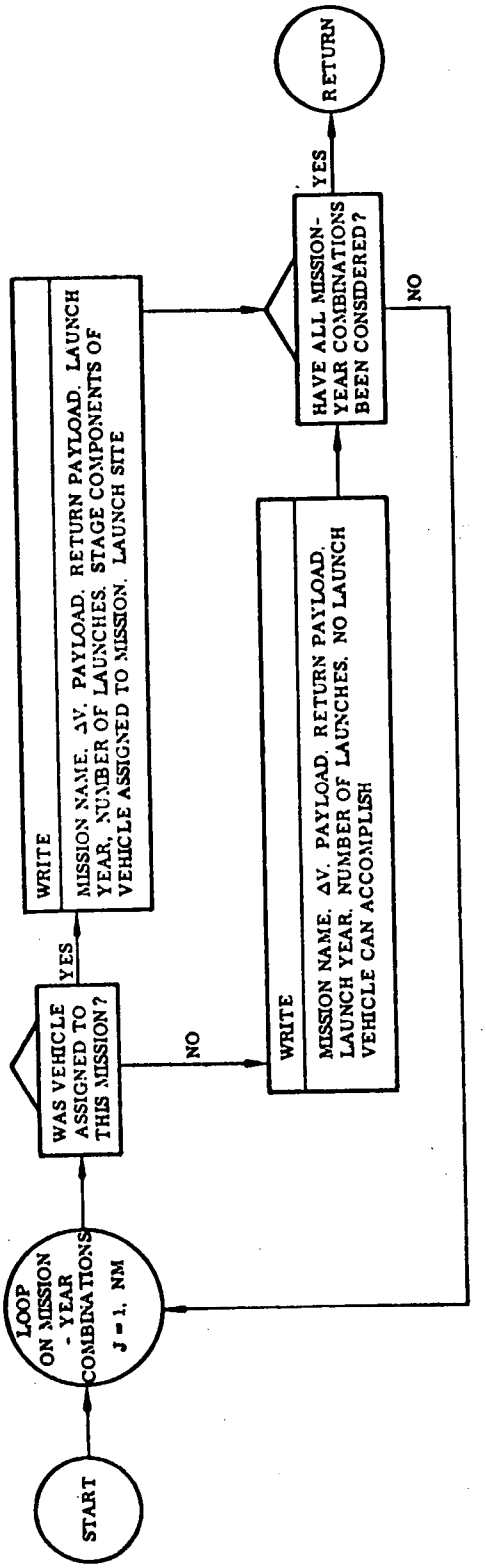
SUBROUTINE NDTR (X, P, D)



SUBROUTINE NDTRI (P, X, C, IE)



SUBROUTINE OUTPTI



SUBROUTINE PACK

IDENTIFICATION

Subroutine PACK
Deck Name MOX01PK
Fortran IV subroutine coded in 360 Assembler Language (also COMPASS coded for the CDC). Written by R. E. Slye

PURPOSE

This subroutine is used to pack an array of integer or logical data into a smaller array in a packed binary format.

METHOD

The unpacked (source) data is treated as an array of unsigned integers. The integer words are truncated on the left and only the N low order bits are retained. The N low order bits are then placed sequentially, left adjusted, in a packed array word until that word is filled. Packing then continues into the next word, etc., until the source data is exhausted.

Since a storage word contains 32 bits, a packed word may contain 32/N data items. Note that since only the N low order bits are retained, the largest integer item that will be represented correctly is $2^N - 1$. For example, if N = 4, the packed items will represent digits from 0 to 15. For a larger integer, the packed item will in effect be the modulus of the source item.

USAGE

This subroutine has three entry points. The three entries are PACK, UNPACK, and ITEM. To pack data, the Fortran call statement is

CALL PACK (L, M, I, N)

where

L is the name of the array containing the source data.
M is the name of the array containing the packed data.
I is the number of data items in L.
N is the number of low order bits to be retained.

The array L should be dimentioned I.
The array M should be dimentioned $\lfloor (I-1)/[32/N] \rfloor + 1$, where [ ] denotes integer part
To unpack data, the Fortran call statement is

CALL UNPACK (L, M, I, N)

where the arguments are as listed above.

I may be less than the actual number of items in the packed array.

Packed data in the array M is unpacked and placed right adjusted in the array L. (The unused high order part of the word is cleared.)

The third entry point to the routine may be used to recover a single item from the packed array M. It is called by the Fortran statement

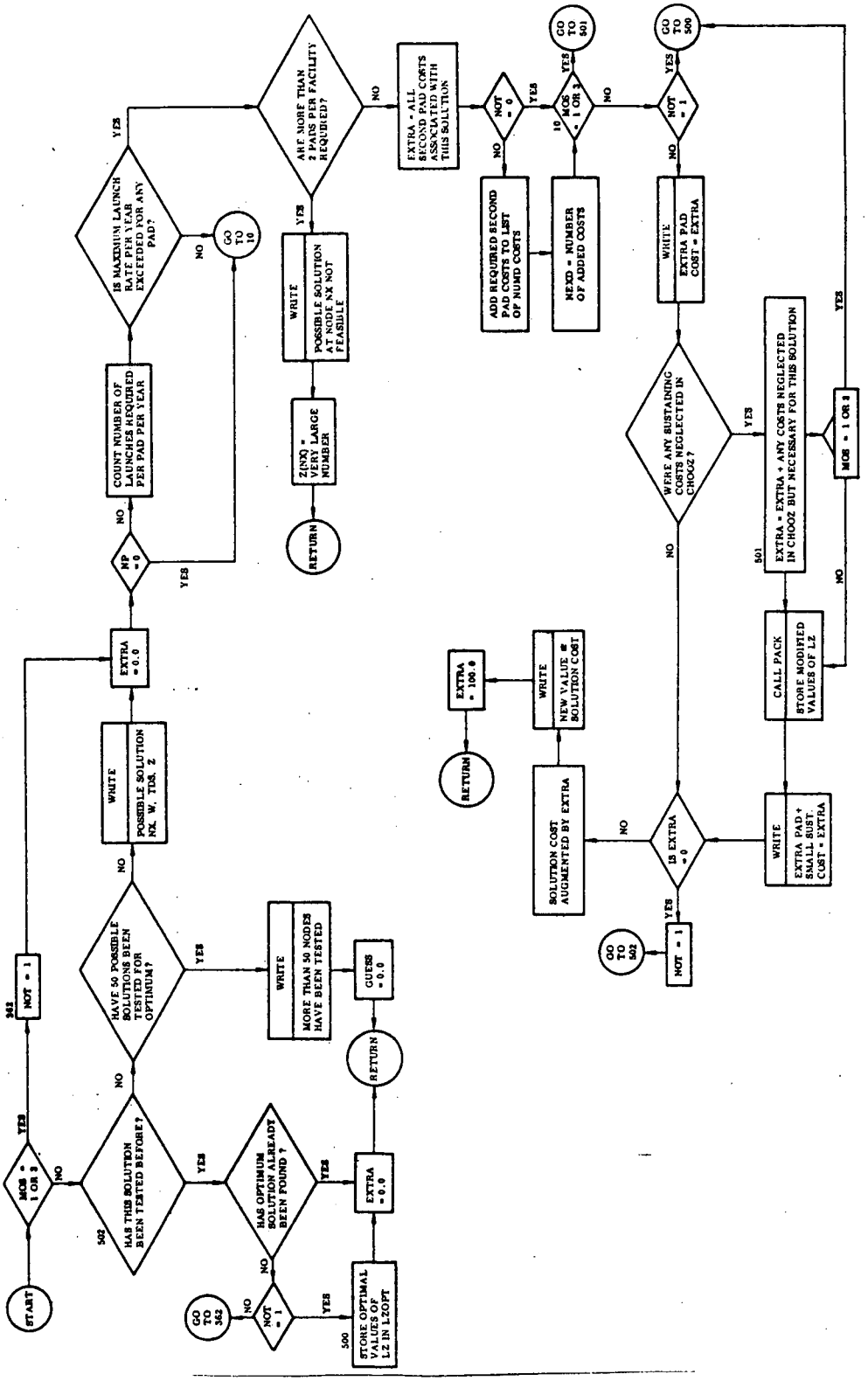
J = ITEM (M, I, N)

The Ith item in the packed array M is returned to the calling program.

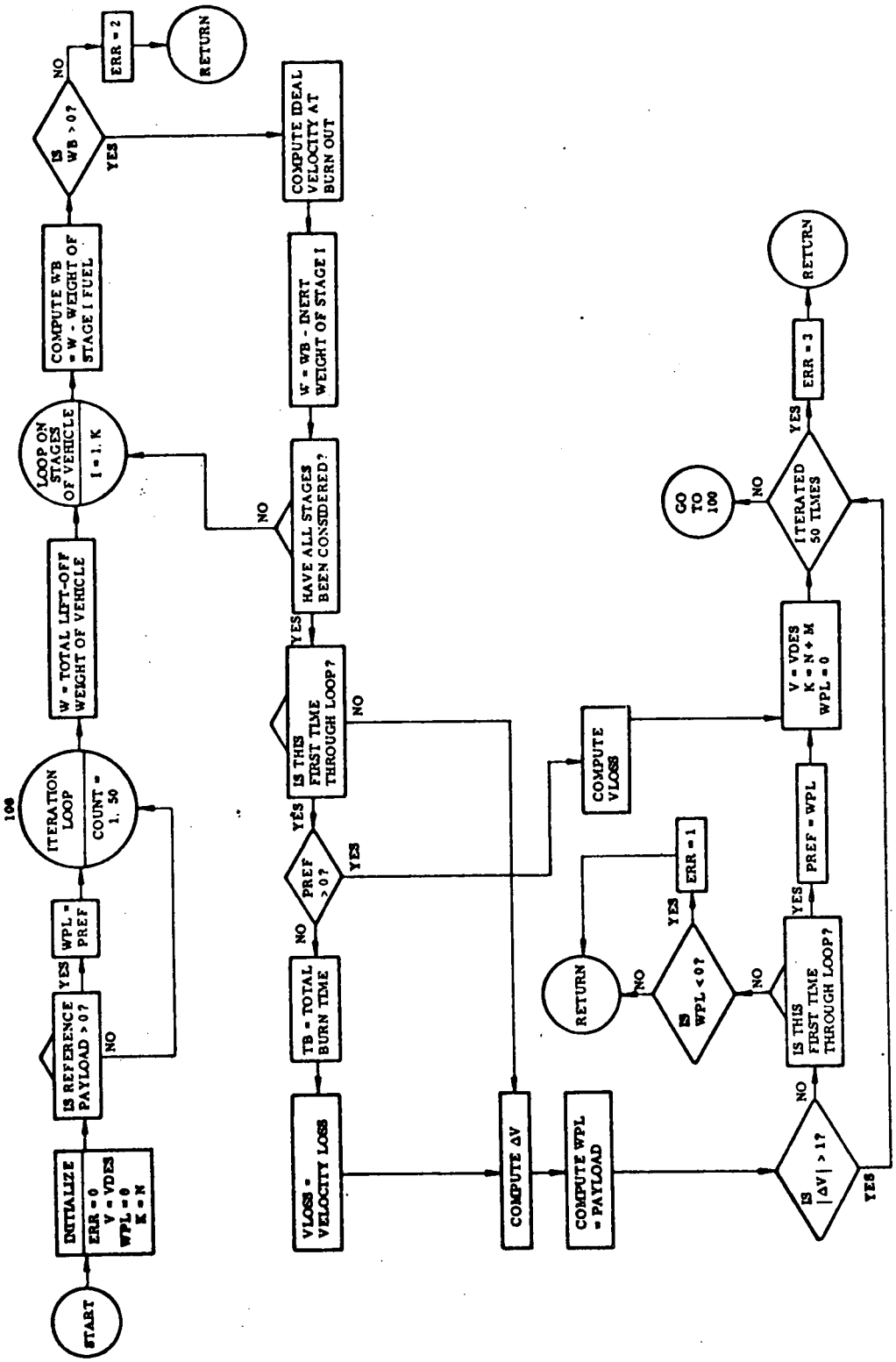
ADDITIONAL INFORMATION

If [32/N] is not an even integer, some low order bits in a packed word are unused. For example, if N = 6 the word may contain 5 items and the last 2 bits are unused. The 6th item will then start at the beginning of the 2nd word.

SUBROUTINE PDCSTI



SUBROUTINE PERFI



SUBROUTINE PLOT

IDENTIFICATION

UMPLOT, Drawing of Graphs by Use of the Printer
360/Assembler Language
Ames Modification of SHARE Library Routine UM PLOT

PURPOSE

This subroutine is used for the purpose of drawing plots, along with the printing of the usual type of numerical output, by use of the printer.

PRELIMINARY REMARKS

Several changes have been incorporated in the FORTRAN IV version of UM PLOT. The maximum width of the plot has been increased from 101 columns to 119 columns. The original program included entries for use in SAP and MAD coded routines, whereas the present version may be entered only from FORTRAN IV or MAP coded programs.

METHOD

A region of core is treated here much as a piece of graph paper. This region of core is called the "image region." The image region is cleared, and then a grid, consisting of I's and -'s, with +.s at grid intersection points, is formed. The program will place any given BCD character at the appropriate place in the image region, corresponding to an ordinate – abscissa pair. Each point is written in the image region independently of those previously written, and so data to be plotted need not be sorted. Any number of points (consistent with the specified size of the image) may be plotted, with any Hollerith plotting character whatever. Points which fall on previously plotted points replace the latter, and points which fall on a grid line replace the grid line character.

Points which lie outside of the specified grid limits are not plotted. When all desired points have been placed in the image region, the latter is written out onto a standard BCD tape (i.e., tape 6, 7, 9, or 11) for subsequent printing.

USAGE

This subroutine has four main entries and two auxiliary entries. The four main entires are PLOT 1, PLOT 2, PLOT 3, and PLOT 4. Each performs a specific function, and normally they are taken in the order listed above. Exceptions to the normal sequence are discussed below. The two auxiliary entries are OMIT and PLTAPE. The first of these is used for the purpose of causing portions of the grid to be deleted, and the second is used if it is desired to output on a tape other than logical tape 6.

Each of the entries is discussed below in detail, following which the calling sequence arguments are defined. It may be noted that the four main entries can be taken by use of either a standard CALL statement [e.g., CALL PLOT 1( )] or an arithmetic statement [e.g., R = PLOT1( )]. The advantage of the latter is that if certain error conditions arise, they can be detected by interrogation of R, whereas the programmer has no way to detect an error condition if the CALL type entry is used. The details concerning error conditions and the interrogation of R will be found in Section D to follow.

A. <u>The Four Main Entries</u>

CALL PLOT 1 (NSCALE, NHL, NSBH, NBL, NSBV)

or

R = PLOT 1 (NSCALE, NHL, NSBH, NBL, NSBV)

This entry is used to set up grid spacing and the total length and width of the graph. The location of decimal points, and the scale factors (powers of 10) for values of the ordinate and abscissa to be printed along the axes of the plot are also specified. If both standard grid and standard scale factors are desired (to be described subsequently), then this entry need not be taken. If several plots are to be printed, all having the same scale factors and grid specifications, then this entry need only be taken one time.

CALL PLOT 2 (IMAGE, XMAX, XMIN, YMAX, YMIN, IDIM)

or

R = PLOT 2 (IMAGE, XMAX, XMIN, YMAX, YMIN, IDIM)

This entry clears the image region and prepares the grid lines of I's and -'s, with +'s at grid line intersection points. It establishes internally formula for computing the location in the image region that corresponds to a given abscissa – ordinate (X_i, Y_i) pair, based on maximum and minimum values as entered through the calling sequence.

CALL PLOT 3 (BCD, X, Y, NDATA)

or

R = PLOT 3(BCD, X, Y, NDATA)

This entry causes a specified Hollerith plotting character to be placed in the appropriate place in the image region for each of the abscissa – ordinate pairs, which are stored in arrays X and Y. This entry may not be taken unless entry PLOT 2 has been taken previously. This entry may be taken repeatedly, if desired, in order to write several sets of data in the image region before it is read out on tape.

CALL PLOT 4 (NCHAR, LABEL)

or

R = PLOT 4(NCHAR, LABEL)

This entry causes the contents of the image region to be written out on logical tape 6 (unless a different tape has been specified by use of the entry PLTAPE, discussed later). The topmost line of the graph will appear one space below the last line previously printed. The ordinate label is specified, and it will appear to the left of the graph. Abscissa labels may be printed above or below the graph by use of standard printout statments. The entry PLOT 4 can be taken repeatedly to obtain several copies of the same graph, if desired. The entry PLOT 2 must have been taken at least once prior to the entry PLOT 4. It is permissible to alter a graph (in the image region) by use of the entry PLOT 3 and then print the result using PLOT 4, without returning to the entry PLOT 2.

B. The Arguments For The Four Main Entries Are Described Here

Note that certain of them may be either integers or floating point quantities, as for example NHL (integer) or HL (floating equivalent of NHL).

NSCALE is an array of dimension 5 that supplies the subroutine with grid and scale factor information

NSCALE(1) = 0, standard grid and scale factors (see note (a), to follow)
≠ 0, grid and scale factors are as defined in NSCALE (2) - NSCALE (5)

NSCALE(2) = I, scale factor such that printed values of the ordinate are 10^I times the actual values

NSCALE(3) = J, J digits will appear to the right of the decimal point in printed ordinate values (J < 8)

NSCALE(4) = K, scale factor such that printed values of the abscissa are 10^K times the actual values

NSCALE(5) = M, M digits will appear to the right of the decimal point in printed abscissa values ($M < 8$)

NHL (or HL) is the number of horizontal grid lines ($NHL > 0$)

NSBH (or BH) is the number of spaces between horizontal grid lines ($NSBH > 0$)

NVL (or VL) is the number of vertical grid lines ($NVL > 0$)

NSBV (or SBV) is the number of spaces between vertical grid lines ($NSBV > 0$, and $NSVB*NVL \leq 119$)

Note (a). Standard scale factors correspond to values of I, J, K, and M of 0, 3, 0, 3, respectively. A standard grid is available which is 101 columns wide starting at column 13, and 51 lines long. It has 10 vertical grid lines and 5 horizontal grid lines, with 10 spaces between both horizontal and vertical grid lines. If both the standard scale factors and standard grid are desired, then the PLOT 1 entry need not be taken. It should be noted, however, that if PLOT 1 has been entered for the purpose of setting up nonstandard conditions, then the latter prevail until PLOT 1 is reentered with different arguments.

Any combination of vertical and horizontal grid lines may be specified, but the vertical grid always starts at column 13. It may extend as far to the right as column 132. The length of the grid is limited only by the dimensions of the image region in core.

Note (b). Integers are printed for the ordinate and/or abscissa scales if $J \leq -1$ and/or $M \leq -1$.

Note (c). If a scale factor is such that overflow or underflow would occur, then the scale factor is treated as zero. The subroutine may shift abscissa scale printout in order to accommodate all of the desired numbers. If the value of an ordinate or abscissa is too large to be printed in the allowed space to the left of the graph it will be truncated from the left.

IMAGE (or AIMAGE) is an array, dimensioned IDIM, which is used as the image region by the subroutine

XMAX is the value of the abscissa at the rightmost grid line

XMIN is the value of the abscissa at the leftmost grid line (XMIN < XMAX)

YMAX is the value of the ordinate at the uppermost grid line

YMIN is the value of the ordinate at the lowermost grid line (YMIN < YMAX)

IDIM is the dimension of the array IMAGE, where IDIM = N*(NSBH*NHL+1) and

$$N = \frac{K}{6} \text{ rounded up for the IBM 7094, or}$$

$$N = \frac{K}{4} \text{ rounded up for the IBM 360}$$

and where

$$K = NSBV*NVL+1$$

(The square brackets in the formula for N signify "integral value.")

Note (d). Set IDIM equal to at least 867 for the standard grid. (1326 for 360).

BCD is the Hollerith plotting character, any character whatever (see note (e), to follow)

X is the array (or single location) that contains the abscissa of the points to be plotted

Y is the array (or single location) that contains the ordinates of the points to be plotted

NDATA (or DATA) is the number of points to be plotted (NDATA > 0)

Note (e). The plotting character may be loaded into cell BCD by use of a DATA statement, that is,

```
DATA BCD/1H*/
```

or, alternatively, it may be entered as a Hollerith literal in the PLOT 3 entry statement, for example,

```
CALL PLOT 3 (1H*, X, Y, NDATA)
```

(The arithmetic statement entry R = PLOT 3 ( ) may not be used in the latter case.)

Note (f). If it is desired to write a single point at a time into the image array, set NDATA equal to 1.

N CHAR (or CHAR) is the number of Hollerith characters, including blanks, in the ordinate label (N CHAR $\leq$ NHL*NSBH+1)

LABEL is an array which contains the Hollerith characters that constitute the ordinate lable to be printed along the leftmost grid line. (See note (g), below)

Note (g). The ordinate label can be entered in array LABEL by use of the DATA statement, that is,

```
DATA (LABEL (J), J = 1, 3)/17HbbbORDINATEbLABEL/
```

Alternatively, it can be loaded as a Hollerith literal in the PLOT 4 entry statement, for example,

```
CALL PLOT 4 (17, 17HbbbORDINATEbLABEL)
```

(The arithmetic statement entry, R = PLOT 4 ( ), may not be used in the latter case.)

R (See Section D, to follow)

C. The Two Auxiliary Entries and Their Arguments

CALL PLTAPE (NTAPE)

This entry is used, prior to PLOT 4, if it is desired that the output be on a tape other than tape 6. Here, NTAPE is the tape number upon which the output is to take place (7, 9, or 11). The output tape number remains as set by this entry until PLTAPE is called again with a different value for NTAPE.

CALL OMIT (NARG)

This entry causes certain portions of the graph to be deleted. It is taken prior to the entry PLOT 4. The settings for NARG are tabulated below

NARG	Effect
1	Numerical values of the abscissa are not printed
2	Numerical values of the ordinate are not printed
3	Combines the effect of NARG = 1 and NARG = 2
4	The complete bottom horizontal grid line is deleted
5	Combines the effect of NARG = 1 and NARG = 4
6	Combines the effect of NARG = 2 and NARG = 4
7	Combines the effect of NARG = 1 NARG = 2, and NARG = 4

D. Error Conditions

If arguments are incompatible with certain restrictions, then the message

$$\text{IMPROPER ARGUMENT} \begin{cases} \text{PLOT 1, or} \\ \text{PLOT 2,} \\ \text{etc.} \end{cases}$$

is printed, thus indicating the entry where the improper entry appears. If such errors occur in PLOT 1 or PLOT 2, subsequent entries into PLOT 3 and PLOT 4 are deleted with no further comment. The argument restrictions are

$$\begin{aligned} &NHL > 0 \\ &NSBH > 0 \\ &NVL > 0 \\ &NSBV > 0 \\ &NSBV * NVL \leq 119 \\ &XMAX > XMIN \\ &YMAX > YMIN \end{aligned}$$

BCD must be a single left-adjusted Hollerith character

If the user attempts to execute PLOT 3 or PLOT 4 without having previously executed PLOT 2, (or without execution of PLOT 2 subsequent to the execution of PLOT 1), the comment

NO PREVIOUS PLOT 2

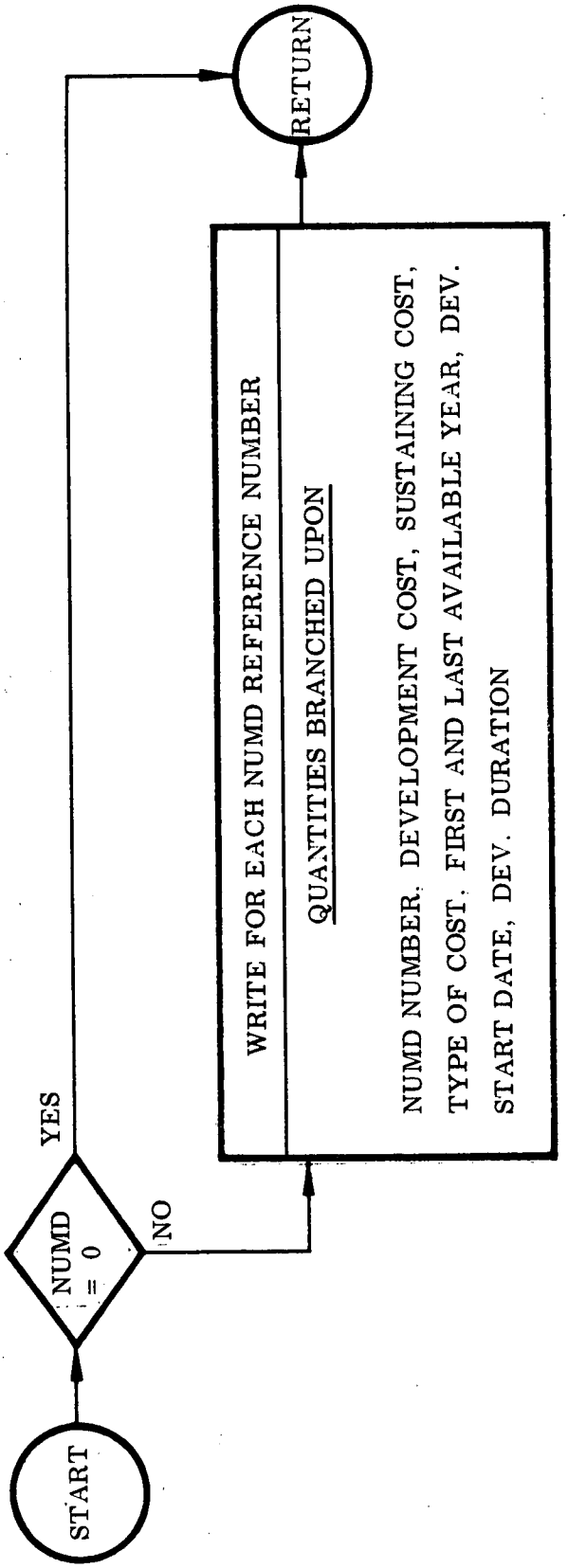
will be printed.

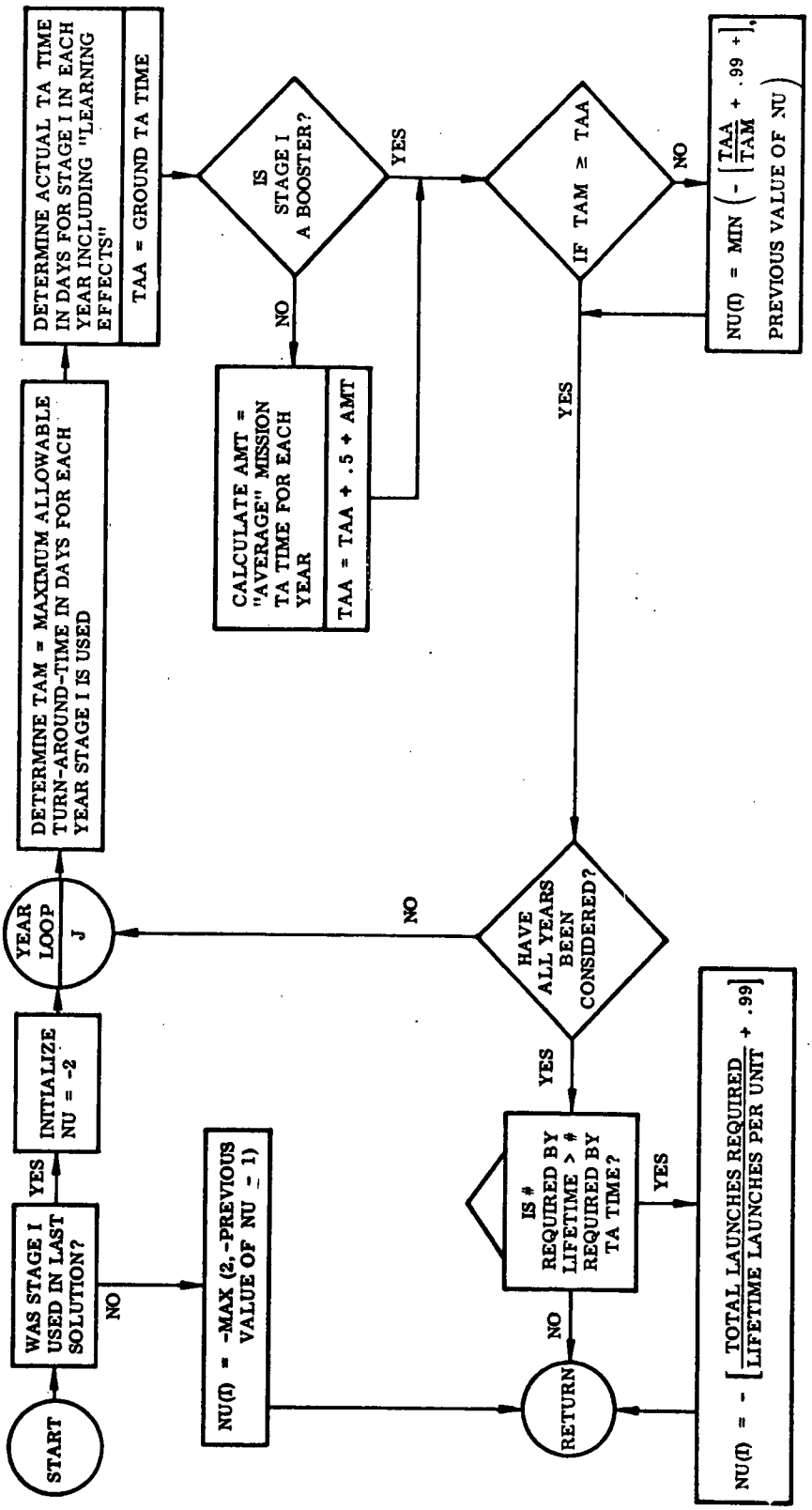
If the arithmetic statement (rather than the CALL statement) is used for the four main entries, then the user may take appropriate action in the case of such errors as would lead to the printouts described above. An error in the arguments, or one due to the

unsuccessful completion of an earlier entry, will cause a + 1.0, + 2.0, + 3.0, or + 4.0 to be loaded in cell R for entries PLOT 1, PLOT 2, PLOT 3, or PLOT 4, respectively. Cell R contains + 0.0 if no error condition arises. The user simply tests R following each attempt to enter the subroutine via PLOT 1, PLOT 2, PLOT 3, or PLOT 4.

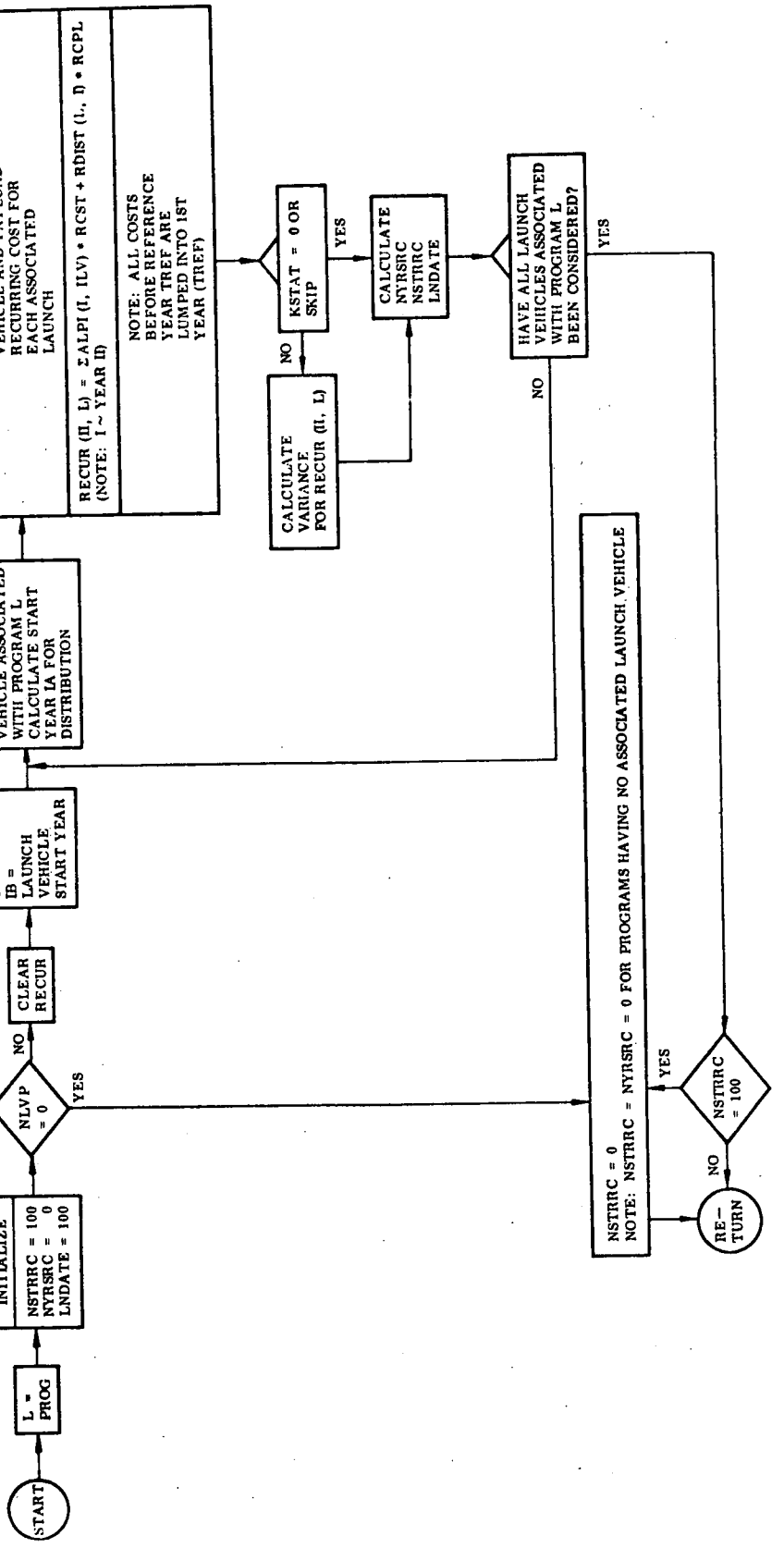
If any points are not plotted by PLOT 3, then the number - 3.0 will be found in R . This might arise if points lie outside the stated minimum and maximum limits of the ordinate and abscissa, and need not be considered an error.

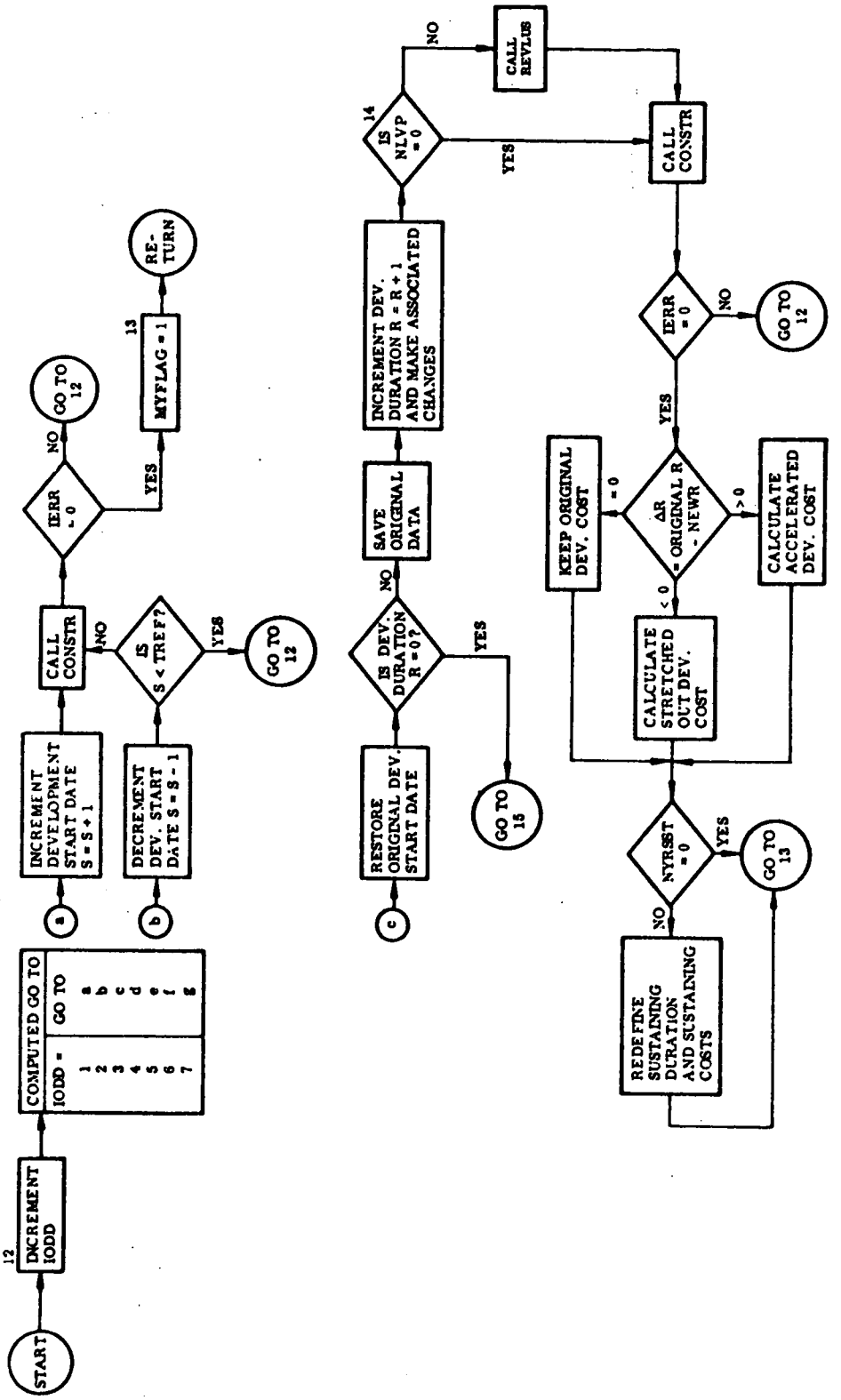
SUBROUTINE PRINTI

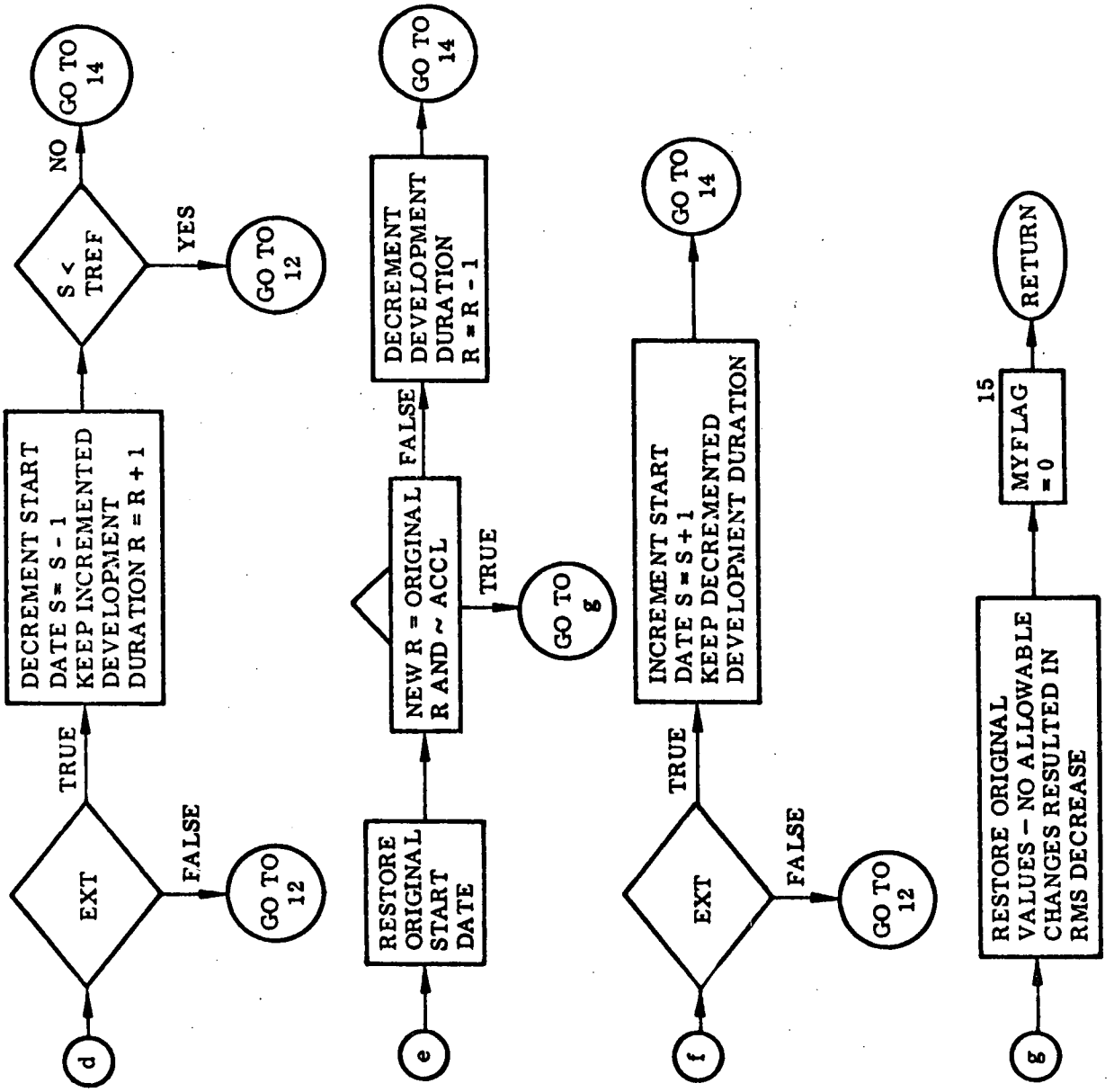



START
WAS STAGE I USED IN LAST SOLUTION?
YES
NO
INITIALIZE NU = -2
NU(I) = -MAX (2, -PREVIOUS VALUE OF NU - 1)
YEAR LOOP J
DETERMINE TAM = MAXIMUM ALLOWABLE TURN-AROUND-TIME IN DAYS FOR EACH YEAR STAGE I IS USED
DETERMINE ACTUAL TA TIME IN DAYS FOR STAGE I IN EACH YEAR INCLUDING "LEARNING EFFECTS"
TAA = GROUND TA TIME
IS STAGE I A BOOSTER?
YES
NO
CALCULATE AMT = "AVERAGE" MISSION TA TIME FOR EACH YEAR
TAA = TAA + .5 * AMT
IF TAM ≥ TAA
NO
YES
NU(I) = MIN (- ⌊TAA/TAM + .99⌋, PREVIOUS VALUE OF NU)
HAVE ALL YEARS BEEN CONSIDERED?
NO
YES
IS # REQUIRED BY LIFETIME > # REQUIRED BY TA TIME?
YES
NO
NU(I) = - ⌊TOTAL LAUNCHES REQUIRED / LIFETIME LAUNCHES PER UNIT + .99⌋
RETURN

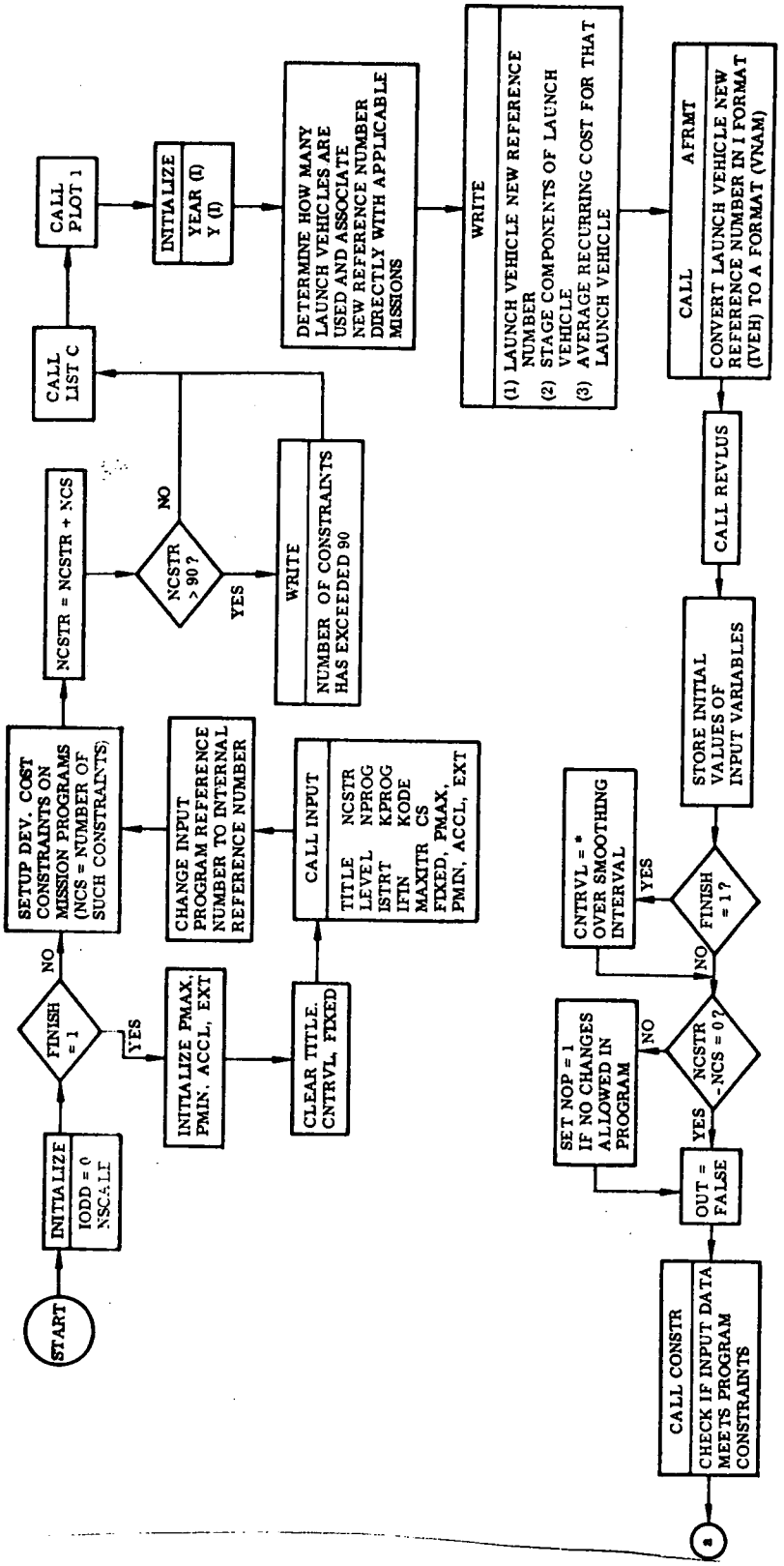
SUBROUTINE REVLUS

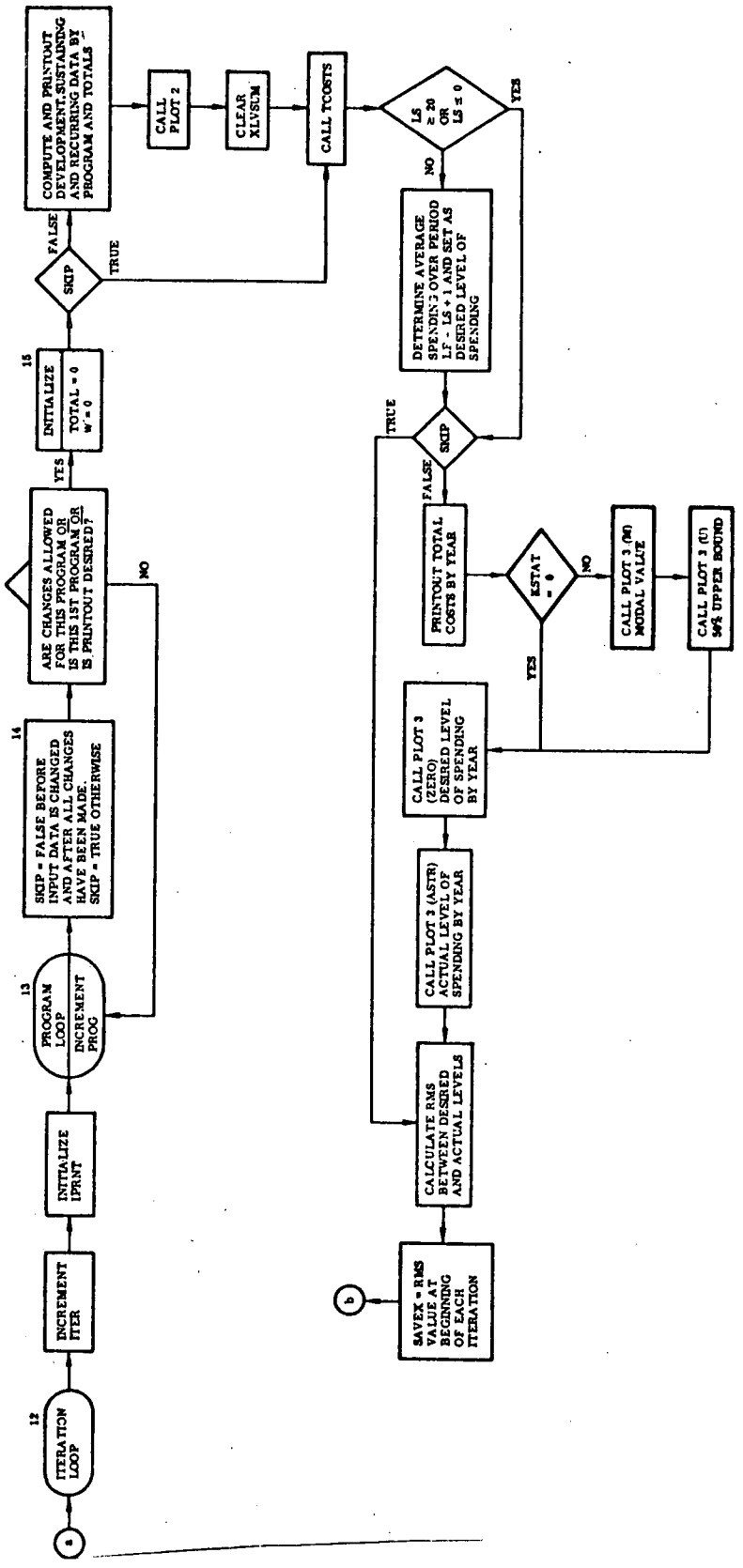


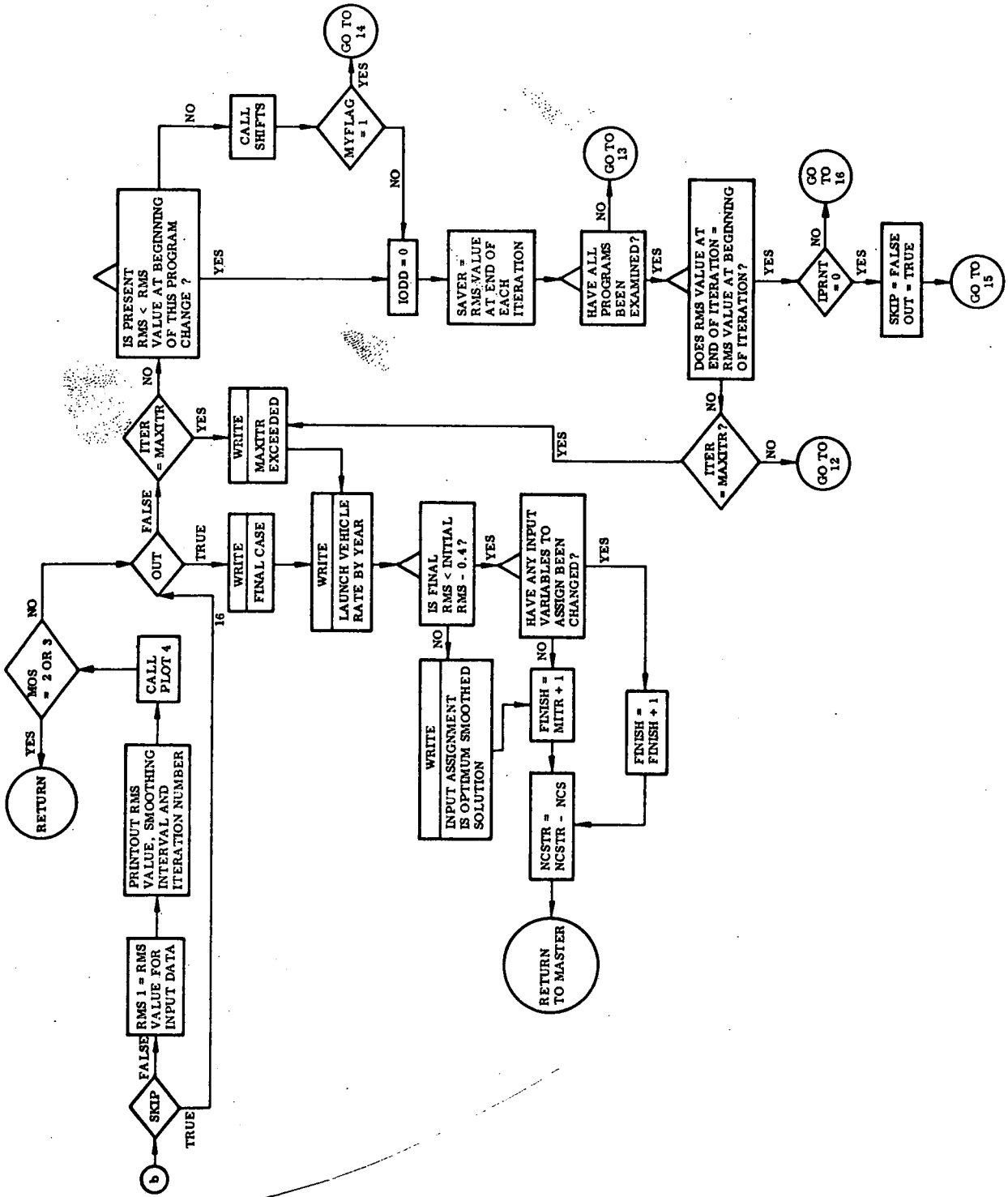

START
12
INCREMENT IODD
COMPUTED GO TO
IODD =
GO TO
1 a
2 b
3 c
4 d
5 e
6 f
7 g
a
INCREMENT DEVELOPMENT START DATE S = S + 1
CALL CONSTR
NO
IS S < TREF?
YES
GO TO 12
b
DECREMENT DEV. START DATE S = S - 1
IERR = 0
NO
GO TO 12
YES
13
MYFLAG = 1
RE-TURN
c
RESTORE ORIGINAL DEV. START DATE
IS DEV. DURATION R = 0?
NO
YES
GO TO 15
SAVE ORIGINAL DATA
INCREMENT DEV. DURATION R = R + 1 AND MAKE ASSOCIATED CHANGES
14
IS NLVP = 0
NO
CALL REVLIS
YES
CALL CONSTR
IERR = 0
NO
GO TO 12
YES
ΔR = ORIGINAL R - NEWR
= 0
KEEP ORIGINAL DEV. COST
< 0
CALCULATE STRETCHED OUT DEV. COST
> 0
CALCULATE ACCELERATED DEV. COST
NYRSST = 0
YES
GO TO 13
NO
REDEFINE SUSTAINING DURATION AND SUSTAINING COSTS


d
EXT
TRUE
FALSE
GO TO 12
DECREMENT START DATE S = S - 1
KEEP INCREMENTED DEVELOPMENT DURATION R = R + 1
S < TREF
YES
GO TO 12
NO
GO TO 14
e
RESTORE ORIGINAL START DATE
NEW R = ORIGINAL R AND ~ ACCL
TRUE
GO TO g
FALSE
DECREMENT DEVELOPMENT DURATION R = R - 1
GO TO 14
f
EXT
TRUE
FALSE
GO TO 12
INCREMENT START DATE S = S + 1
KEEP DECREMENTED DEVELOPMENT DURATION
GO TO 14
g
RESTORE ORIGINAL VALUES – NO ALLOWABLE CHANGES RESULTED IN RMS DECREASE
15
MYFLAG = 0
RETURN

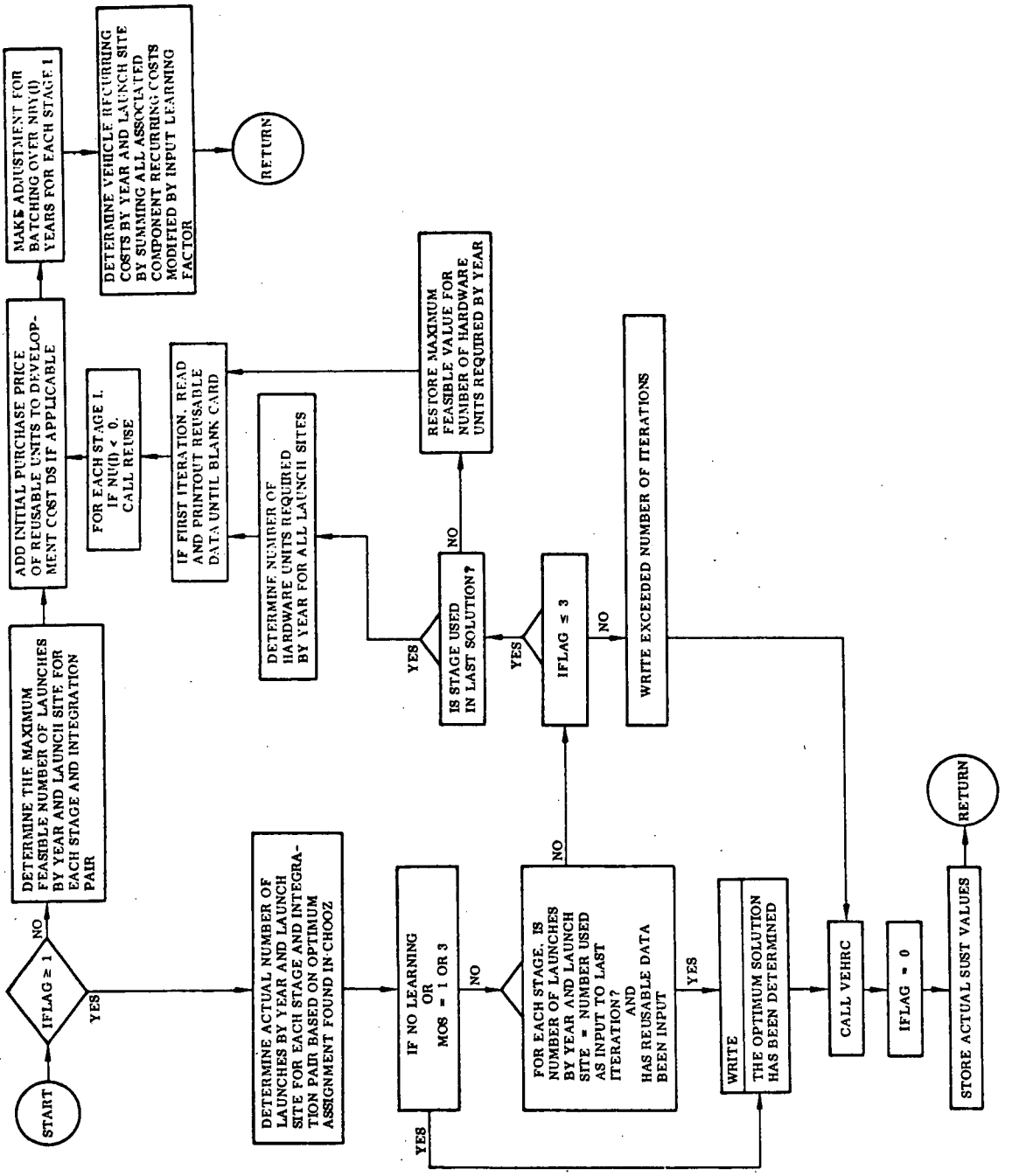
SUBROUTINE SMOOTH



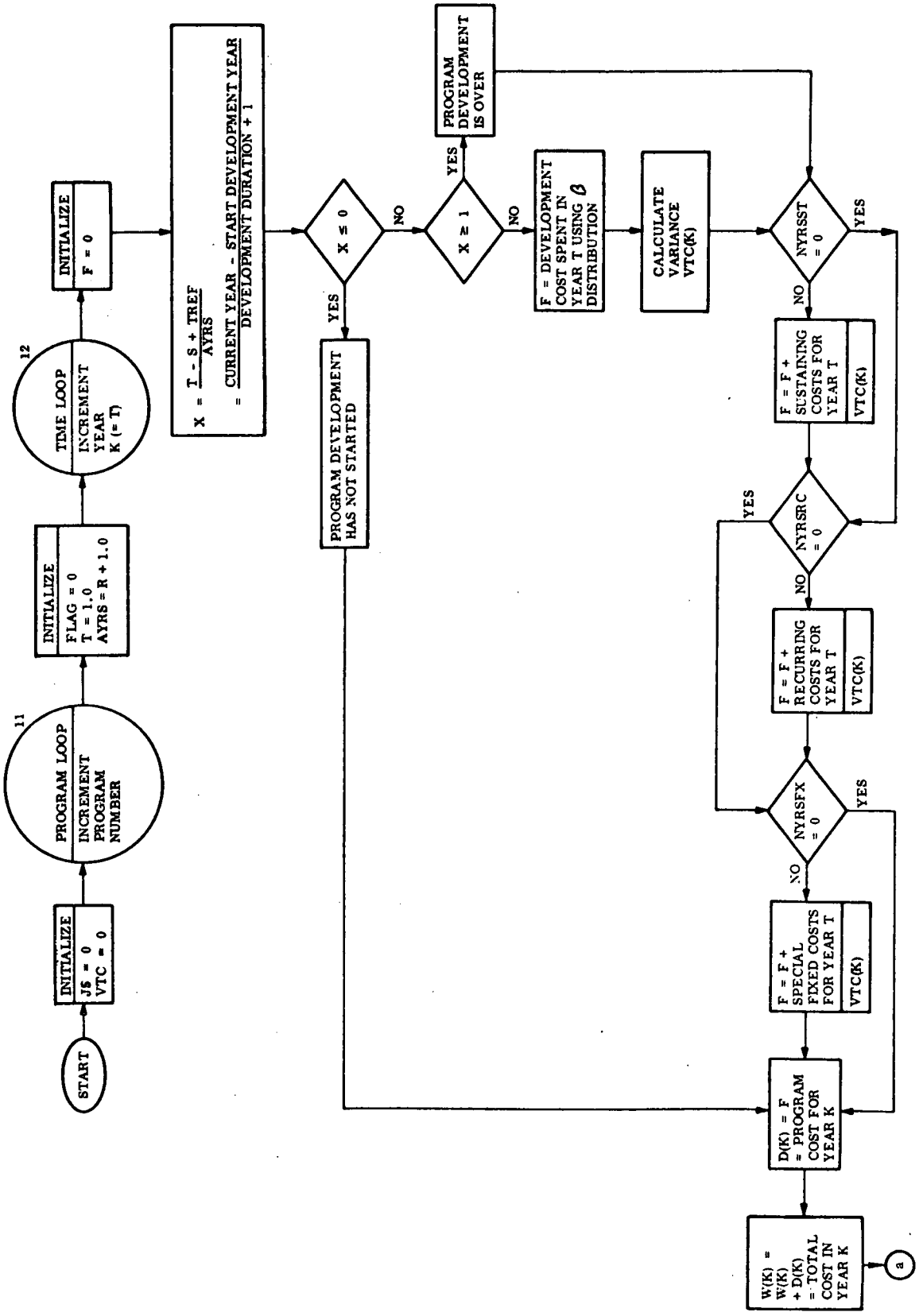

a
12
ITERATION LOOP
INCREMENT ITER
INITIALIZE IPRINT
13
PROGRAM LOOP
INCREMENT PROG
SKIP = FALSE BEFORE INPUT DATA IS CHANGED AND AFTER ALL CHANGES HAVE BEEN MADE. SKIP = TRUE OTHERWISE
14
ARE CHANGES ALLOWED FOR THIS PROGRAM OR IS THIS 1ST PROGRAM OR IS PRINTOUT DESIRED?
YES
NO
15
INITIALIZE
TOTAL = 0
W = 0
SKIP
FALSE
TRUE
COMPUTE AND PRINTOUT DEVELOPMENT, SUSTAINING AND RECURRING DATA BY PROGRAM AND TOTALS
CALL PLOT 2
CLEAR XLVSUM
CALL TCOSTS
LS ≥ 20 OR LS ≤ 0
YES
NO
DETERMINE AVERAGE SPENDING OVER PERIOD LF - LS + 1 AND SET AS DESIRED LEVEL OF SPENDING
SKIP
TRUE
FALSE
PRINTOUT TOTAL COSTS BY YEAR
KSTAT = 0
NO
YES
CALL PLOT 3 (M) MODAL VALUE
CALL PLOT 3 (U) 90% UPPER BOUND
CALL PLOT 3 (ZERO) DESIRED LEVEL OF SPENDING BY YEAR
CALL PLOT 3 (ASTR) ACTUAL LEVEL OF SPENDING BY YEAR
CALCULATE RMS BETWEEN DESIRED AND ACTUAL LEVELS
b
SAVEX = RMS VALUE AT BEGINNING OF EACH ITERATION

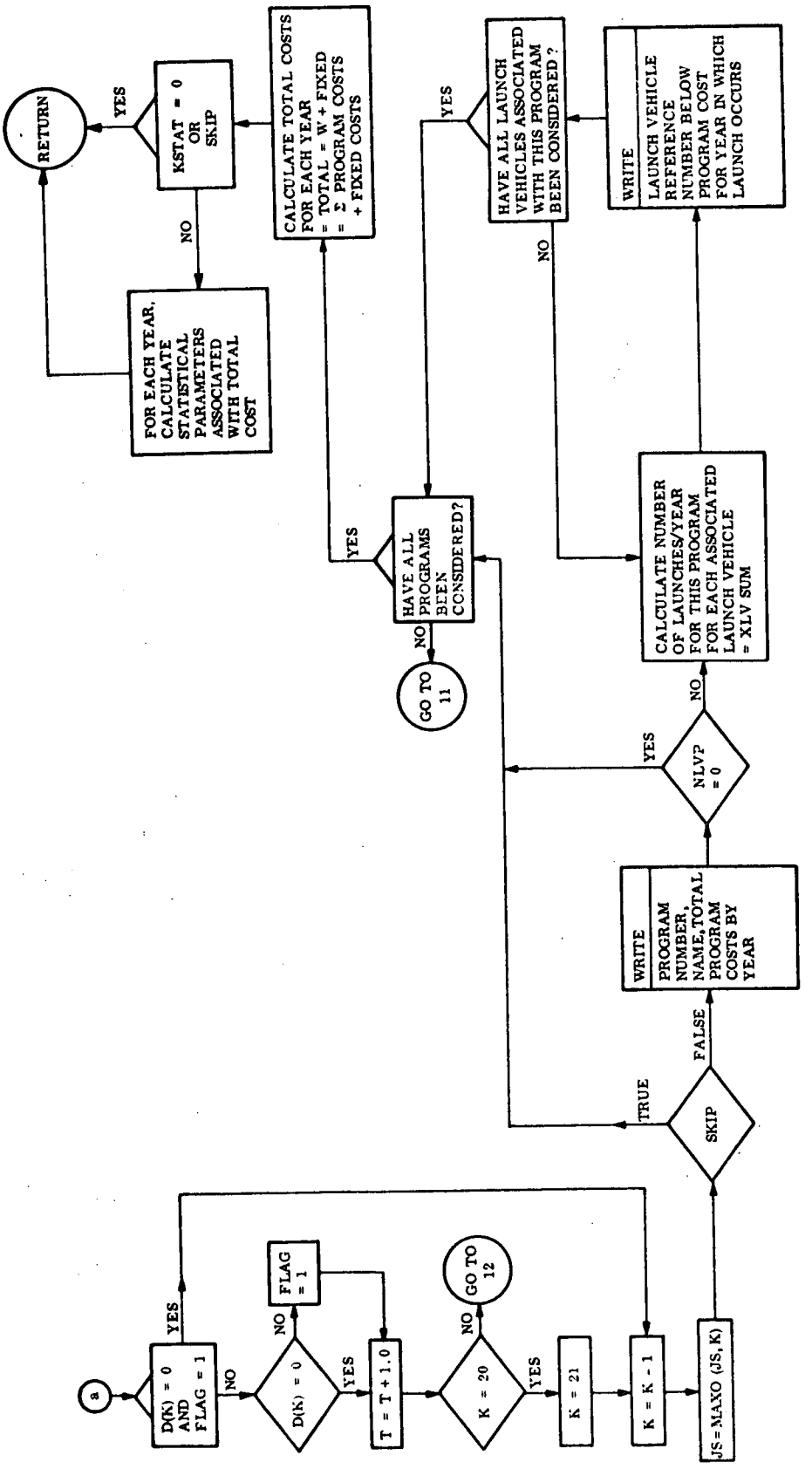

RETURN
MOS = 2 OR 3
NO
YES
CALL PLOT 4
PRINTOUT RMS VALUE, SMOOTHING INTERVAL AND ITERATION NUMBER
RMS 1 = RMS VALUE FOR INPUT DATA
SKIP
FALSE
TRUE
b
OUT
FALSE
TRUE
16
WRITE FINAL CASE
WRITE LAUNCH VEHICLE RATE BY YEAR
IS FINAL RMS < INITIAL RMS - 0.4?
YES
NO
WRITE INPUT ASSIGNMENT IS OPTIMUM SMOOTHED SOLUTION
HAVE ANY INPUT VARIABLES TO ASSIGN BEEN CHANGED?
YES
NO
FINISH = MITR + 1
FINISH = FINISH + 1
NCSTR = NCSTR - NCS
RETURN TO MASTER
ITER = MAXITR
YES
NO
WRITE MAXITR EXCEEDED
IS PRESENT RMS < RMS VALUE AT BEGINNING OF THIS PROGRAM CHANGE ?
YES
NO
CALL SHIFTS
MYFLAG = 1
YES
NO
GO TO 14
IODD = 0
SAVER = RMS VALUE AT END OF EACH ITERATION
HAVE ALL PROGRAMS BEEN EXAMINED?
NO
YES
GO TO 13
DOES RMS VALUE AT END OF ITERATION = RMS VALUE AT BEGINNING OF ITERATION?
YES
NO
IPRNT = 0
YES
NO
GO TO 16
SKIP = FALSE OUT = TRUE
GO TO 15
ITER = MAXITR?
NO
YES
GO TO 12

SUBROUTINE STGNMI

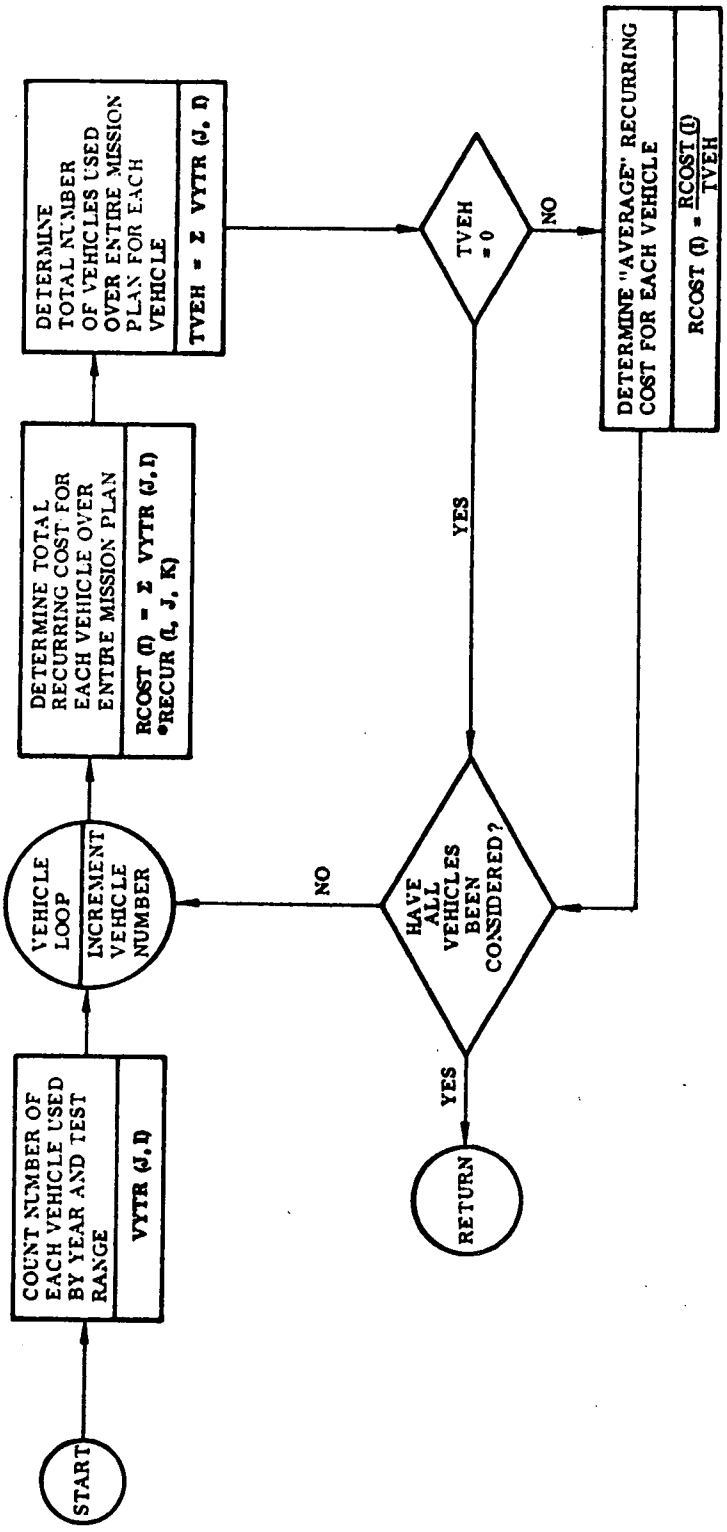


SUBROUTINE TCOSTS




a
D(K) = 0 AND FLAG = 1
YES
NO
D(K) = 0
NO
YES
FLAG = 1
T = T + 1.0
K = 20
NO
YES
GO TO 12
K = 21
K = K - 1
JS = MAXO (JS, K)
SKIP
TRUE
FALSE
WRITE PROGRAM NUMBER, NAME, TOTAL PROGRAM COSTS BY YEAR
NLVP = 0
YES
NO
CALCULATE NUMBER OF LAUNCHES/YEAR FOR THIS PROGRAM FOR EACH ASSOCIATED LAUNCH VEHICLE = XLV SUM
HAVE ALL LAUNCH VEHICLES ASSOCIATED WITH THIS PROGRAM BEEN CONSIDERED ?
YES
NO
WRITE LAUNCH VEHICLE REFERENCE NUMBER BELOW PROGRAM COST FOR YEAR IN WHICH LAUNCH OCCURS
HAVE ALL PROGRAMS BEEN CONSIDERED?
YES
NO
GO TO 11
CALCULATE TOTAL COSTS FOR EACH YEAR = TOTAL = W + FIXED = Σ PROGRAM COSTS + FIXED COSTS
KSTAT = 0 OR SKIP
YES
NO
RETURN
FOR EACH YEAR, CALCULATE STATISTICAL PARAMETERS ASSOCIATED WITH TOTAL COST

SUBROUTINE VEHRC



Appendix D
PROGRAM LISTING

D.1 DESCRIPTION

A compile-and-save Fortran listing of each major subroutine in the optimal assignment/budget smoothing program is included in this section. Storage requirements for each subroutine are listed on the output along with the code name under which the subroutine was saved. Total storage requirements are listed at the beginning of the sample case presented in Appendix B. Comment cards describing the logical function of each subsection and defining any variables whose names are not mnemonic are liberally distributed throughout the deck so that new users may readily become familiar with the programs.

Subroutines INPUT and PLOT are stored for general NASA use. Therefore, no listing is included here; however, a description of each is provided in Appendix C for completeness. Subroutines AFRMT and PACK are written in 360 assembler language, so the listings are provided in that language.

Labeled common blocks were used for storage whenever possible to avoid long argument lists for each subroutine. These blocks are found at the beginning of each listing. Subroutine ASSIGN lists all subroutines in which each common block appears. The block labeled SCRACH stores variables only required in that subroutine or related subroutines, so these storage locations may be used for storing different variables in the next subroutine. All other labeled common blocks contain variables used by the same name in several subroutines.

The listings are presented in alphabetical order according to subroutine name for easy reference. The main subroutine is listed under the name MASTER.

D.2 COMPILE-AND-SAVE LISTING

The compile-and-save listing follows.

```
                                                    EXTERNAL SYMBOL DICTIONARY
SYMBOL   TYPE ID  ADDR   LENGTH LD ID

AFRMT     SD  01 000000 000040
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

000000                                1 AFRMT   CSECT
000000                                2         USING  *,15          REG 15 FOR BASE
000000 5020 D01C              0001C   3         ST     2,28(0,13)    SAVE REG 2
000004 9812 1000              00000   4         LM     1,2,0(1)      LOAD ADDRESSES OF ARGS TO REGS 1-2
000008 5810 1000              00000   5         L      1,0(0,1)      DATA TO REG 1
00000C 4E10 F038              00038   6         CVD    1,WORK        CONVERT TO DECIMAL
000010 F332 2000 F03D 00000   003D    7         UNPK   0(4,2),WORK+5(3)   UNPACK 4 DIGITS
000016 96F0 2003      00003           8         OI     3(2),X'F0'    INSERT ZONES
00001A 4110 0004              00004   9         LA     1,4
00001E 95F0 2000      00000          10 LOOP    CLI    0(2),C'0'     SCAN OUT LEADING ZEROS
000022 4770 F032              00032  11         BNE    RETURN
000026 9240 2000      00000          12         MVI    0(2),C' '     INSERT BLANK
00002A 4120 2001              00001  13         LA     2,1(0,2)      BUMP POINTER
00002E 4610 F01E              0001E  14         BCT    1,LOOP        LIMIT TO 4 CHARACTERS
000032 5820 D01C              0001C  15 RETURN  L      2,28(0,13)    RESTORE REG 2
000036 07FE                          16         BR     14            RETURN
000038                               17 WORK    DS     D
                                     18         END
```

```
                                                    CROSS-REFERENCE

SYMBOL     LEN  VALUE  DEFN     REFERENCES

AFRMT      00001 000000 0001
LOOP       00004 00001E 0010     0014
RETURN     00004 000032 0015     0011
WORK       00008 000038 0017     0006  0007


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
   32 PRINTED LINES
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000       NAME MUX02AT(R)
****MUX02AT    NOW REPLACED IN DATA SET
```

```
0001            SUBROUTINE ASSIGN
                C
                C     THIS PROGRAM GENERATES THE LEAST COST ASSIGNMENT OF LAUNCH
                C     VEHICLES TO SPACE MISSIONS.  A BRANCH AND BOUND TECHNIQUE IS USED
                C     TO REDUCE THE COMBINATORIAL COMPLEXITY OF THE PROBLEM. SEVERAL BRANCHES
                C     ARE CREATED AT EACH NODE.  ONE OF THE BRANCHES EXCLUDES THE NEXT
                C     COST AND THE OTHERS    ASSUME EXPENDITURE OF A NON-RECURRING COST
                C     WITH 1-2 YEARS OF SUSTANING COST ADDED AT EACH NODE.
                C     PENALTY FUNCTIONS ARE USED TO SHARPEN THE LOWER BOUND.
                C ****THIS VERSION USES RATE EFFECTS IN RECURRING COSTS****
                C*****THIS VERSION INCORPORATES PAD COSTS AND REUSABLE PARAMETERS******
                C
0002                DOUBLE PRECISION NAME
0003                REAL NPERPD
0004                LOGICAL EXT,ACCL
0005                INTEGER*2   YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,KODE,
                   1 NYRSFX,KODEM,KODESP,NU,NBY,MODE,NOB,FINISH,NSTG,NFML,NFMU,KODS,
                   2  MAS,LABS,LABF,LABI,LSA,NYS,KODEF,LST,MST,IST,JST,KST,VEH,NYD,
                   3  NMULT,NONREC,IS,MAT,LYR,LETT,LYD,MIN,NVS,MRV,NRP,NYP,KODEP,
                   4  IVEHA,NTRIP,NPLS,NRR,MR,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,MAPS,
                   5 MAPF,MAPI,KOUT,LTR,KODEV,NINTYR,NTGYTR,MAF,MAIC
                C
                C  STORAGE FOR TCOST, ASSIGN, AND MASTER AND SMOOTH,DATINS
0006                COMMON/SAVER/  RFIXD(12,84)
                C  STORAGE FOR DECISN, MATCH, PRINT AND ASSIGN,DATINS
0007                COMMON/SAVDMP/ NFAM,KFLAG,FAM(30),KODEF(30),FMNR(30),FMSUS(30),
                   1JST(30),YDF(30),LSA(40),SNR(40),NYS(40),DINT(40),SINT(40),KST(40),
                   2  YDI(40),YDS(40),IST(40),FMSLS(30,2),SUSLS(40,2),SINTLS(40,2),
                   3  LST(30,5),YDPF(30,5),MST(30,10),YDPS(30,10)
                C  STORAGE FOR ASSIGN, STGNUM, AND REUSE,DATINS, COMPARE,MASTER
0008                COMMON/SAVSAR/COR,PUC(3),SRJ(3,3),NU(40),NBY(40),NOB(40),RINT(40),
                   1   PLCINT(40),XLT(40), PLCT(40),UPP(40),TAT(40),TAMT(50),SR(40,3),
                   2   MODE(40,3),PLC(40,3)
                C   STORAGE FOR MASTER,ASSIGN,DECISN,STGNUM,SMOOTH,AVAIL,MATCH,PRINT,CAPABL,
                C       AND OUTPUT AND PDCSTI,DATINS, COMPARE,TCOSTS
0009                COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                   1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                   2  RPLM(50),MAS(40,3), RXD(12,50)
                C  STORAGE FOR MASTER,ASSIGN,SMOOTH,TCOST,OUTPUT,SHIFT,CONSTR,DATINS
0010                COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                   1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                   2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                   3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
```

```
                   4,  R(84), S(84),CS(90),NPROG(90),KPROG(90), KODE(90)
                C   STORAGE FOR ASSIGN,CHOOZ,LBOUND,DECISN,PADCST,CAPABL,STGNUM,MASTER,SMOOTH,
                C      PRINT, REVALU,TCOST, VEHREC ,MATCHI,DATINS, COMPARE
0011                COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                   1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
                C   STORAGE FOR ASSIGN,PDCST,CAPABL,DECISN,MATCH,PRINT,STGNUM,DATINS,COMPARE
0012                COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                   *                NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                   1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                   2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                   3   PSTGS(30,10,2)
                C   STORAGE FOR ASSIGN,CAPABL, AND AVAIL,DATINS
0013                COMMON/SVACAV/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
                   1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
                   2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
                C   STORAGE FOR MASTER,CHOOZ,ASSIGN,STGNUM,PADCST,LBOUND,REUSE,VEHREC,
                C   OUTPUT,AVAIL,CAPABL,MATCH,SMOOTH,DECISN,PRINT,DATINS,COMPARE,REVALU,TCOSTS
0014                COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                   1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
                C   TEMPORARY STORAGE FOR ASSIGN,CHOOZ,STGNUM,LBOUND,VEHREC,AVAIL,PADCST
                C  DATINS, COMPARE
0015                COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
                   1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
                C  OVERLAY STORAGE
0016                COMMON/SCRACH/  IP,IV,IG,MODX(3),NFX(4),NPFAX(5),LSX(5),NPINXL(5),
                   1  NPINXU(5),NPSTX(10),MSX(10),LZ(20),PB(50),MISN(50,20),DUM(1382),
                   2  RCOST(60),RXM(50),II,KNSP,KLCK,IM,DUMM(4192)
                C
0017                NEXD = 0
0018                IFLAG = 0
0019                KFLAG = 0
0020                IF(FINISH.GT.1)  GUESS = 1.75*GUESS
0021                IF(FINISH.GT.1)  GO TO 17
0022                KNV = 100
0023               11 NSTG = 0
0024                NFAM = 0
0025                NCI = 0
0026                NMIS = 0
0027                NSPR = 0
0028                NUMD = 1
0029                NP = 0
0030                NV = 0
```

```
FORTRAN IV G LEVEL 1, MOD 4            ASSIGN            DATE = 71312          17/22/36

            C  LCK = LEARNING CURVE KODE; = 1 IF HAVE LEARNING CURVE EFFECTS; =0 IF NONE
0031                LCK = 0
            C
0032                CALL DATINS
            C
0033                IF(MYRS.EQ.0)  RETURN
0034                IF(IM.LT.0)  GO TO 3000
            C
            C     ***SET UP MISSION MATRIX BY YEAR***
0035                NM = 0
0036                DO 4 I = 1,NMIS
0037                DO 4 J=1,MYRS
0038                IF(MISN(I,J).EQ.0) GO TO 4
0039                NM = NM + 1
0040                YRLM(NM)= FLOAT(MISN(I,J)) * PB(I)
0041                LETT(NM)= I
0042                LYR(NM) = J
0043              4 CONTINUE
            C
0044           3000 IF(GUESS.GT.1.0)  GO TO 3005
0045                GUESS = 1.0E15
            C
0046           3005 CALL CAPBLI
            C
0047             16 WRITE(6,2001) NSTG,NV,NFAM,NCI,NP,NMIS,MYRS,ILY,GUESS,NOPT,NSOL,
                   1  GRO, COR
0048                IF(LCK.EQ.0)  GO TO 17
            C       CALCULATE EXPONENT FOR LEARNING CURVE
0049                ALOG2 = ALOG(2.)
0050                IF (IG.LT.0) GO TO 8030
0051                DO 660 I=1,NSTG
0052                DO 660 J=1,3
0053                IF (MODE(I,J).EQ.0.AND.PLC(I,J).GT..001)
                   1PLC(I,J) = ALOG(PLC(I,J))/ALOG2
0054             660 CONTINUE
0055            8030 IF(II.LT.0.OR.NCI.EQ.0)  GO TO 17
0056                DO 680 I=1,NCI
0057                IF(PLCINT(I).GT..001)
                   1PLCINT(I) = ALOG(PLCINT(I))/ALOG2
0058             680 CONTINUE
            C
0059             17 IF(NUMD.EQ.0)  GO TO 305
            C
0060                CALL DECSNI
            C
0061                IF(KFLAG.EQ.1)  GO TO 1
            C
0062             305 CALL AVAILI
            C
            C
0063                CALL STGNMI
            C
0064                GUESS1 = GUESS
            C
0065             620 CALL CHOOZS
            C
0066                IF(NEXT.GE.500.OR.GUESS.LT..001)  GO TO 2
            C         IFLAG = NUMBER OF TIMES CHOOZ HAS BEEN CALLED
0067                IFLAG = IFLAG + 1
            C
0068                CALL STGNMI
            C
0069                GUESS = GUESS1
0070                IF(IFLAG.EQ.0)  GO TO 1
0071                IF(IFLAG.EQ.100)  GO TO 2
0072                GO TO 620
0073              2 MYRS = 100
0074              1 KNSTG = NSTG
0075                KNFAM = NFAM
0076                KNCI = NCI
0077                KNP = NP
0078                KLCK = LCK
0079                KNMIS = NMIS
0080                KNV = NV
0081                KNSP = NSPR
0082                RETURN
0083           2001 FORMAT (17H1NUMBER OF STAGES,8X,I5/19HONUMBER OF VEHICLES,6X,I5/
                   1  19HONUMBER OF FAMILIES,6X,I5/28HONUMBER OF INTEGRATION COSTS,I2/
                   X  24HONUMBER OF PAD COMPLEXES,4X,I2/
                   2  19HONUMBER OF MISSIONS,6X,I5/16HONUMBER OF YEARS,9X,I5/
                   3  17HOLAUNCH BASE YEAR,8X,I5/15HOTOTAL ESTIMATE,F17.2/14HOOPTION N
                  4UMBER,   11X,I5/  20HONUMBER OF SOLUTIONS,5X,I5/ 17HOINFLATION FAC
                  5TOR,12X,F4.3/12HOCORRELATION,17X,F3.2)
0084                END
```

```
FORTRAN IV G LEVEL 1, MOD 4          ASSIGN          DATE = 71312          17/22/36

  TOTAL MEMORY REQUIREMENTS 0008DE BYTES
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                DEFAULT OPTION(S) USED
IEW0000       NAME MOX02AS(R)
IEW0461   DATINS
IEW0461   CAPBLI
IEW0461   IBCOM=
IEW0461   DECSNI
IEW0461   AVAILI
IEW0461   STGNMI
IEW0461   CHOUZS   •
IEW0461   ALOG
```

MODULE MAP

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
ASSIGN	00	8DE						
SAVER	8E0	FC0						
SAVDMP	18A0	14BC						
SAVSAR	2D60	A5C						
SAVE1	37C0	FC4						
SAV2	4788	FE0						
SAV3	5768	980						
SAV4	60E8	3188						
SVACAV	9270	B48						
SAVALL	9DB8	3A1C						
TEMP	D7D8	4110						
SCRACH	118E8	6A60						

```
 ENTRY ADDRESS       00
 TOTAL LENGTH     18348

****MOX02AS   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                DATE  71.312/16.53.56

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NO.
  ISN 0002            SUBROUTINE AVAILI
                C     *** ADD AVAILABILITY TO VEHICLE CAPABILITY MATRIX***
  ISN 0003            REAL NPERPD
  ISN 0004            INTEGER*2 NVS,MRV,NRP,NYP,KODEP,IVEHA,NTRIP,NPLS,NRR,MR,KODEV,
                     1 FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,VEH,NMULT,NONREC,
                     2  NYD,IS,MAT,LYR,LETT,LYD,MIN,LTR
  ISN 0005            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                     1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                     2  RPLM(50),MAS(40,3), RXD(12,50)
  ISN 0006            COMMON/SAV3/GRD,GUESS,LP,NSOL,MSOL,NP,MDS,NMIS,NSPR,NPERPD(30),
                     1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
  ISN 0007            COMMON/SVACAV/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
                     1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
                     2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
  ISN 0008            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                     1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                     2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
  ISN 0009            COMMON/TEMP/ VNM(2,250),IFLAG,LZ(60),DUM(3603)
                C
  ISN 0010            DO 39 J = 1,NM
  ISN 0011            KO = LYR(J)
  ISN 0012            L = LETT(J)
  ISN 0013            DO 35 I = 1,NV
  ISN 0014            LZ(I) = 0
  ISN 0015            IF(ITEM(VM(1,I),L,1).EQ.0)  GO TO 35
  ISN 0017            K = I
  ISN 0018            IF(LTR(L).EQ.2)  K = I & NV
  ISN 0020            DO 36 M = 1,20
  ISN 0021            IF(NONREC(K,M).EQ.0)  GO TO 37
  ISN 0023            NO = NONREC(K,M)
  ISN 0024            IF(KO.LT.NYD(NO))  GO TO 35
  ISN 0026            IF(KO.GT.LYD(NO))  GO TO 35
  ISN 0028         36 CONTINUE
  ISN 0029         37 LZ(I) = 1
  ISN 0030         35 CONTINUE
  ISN 0031            CALL PACK(LZ,VNM(1,J),NV ,1)
  ISN 0032         39 CONTINUE
                C     NM LESS THAN 181 FOR PRESENT FORMATS
  ISN 0033        284 WRITE (6,4000)
  ISN 0034            DO 421 ITER = 1,4
  ISN 0035            KNM = MINO(ITER*45,NM)
  ISN 0036            K = 1 & (ITER - 1)*45

  ISN 0037        285 WRITE(6,4002)  (LETT(J), J = K,KNM)
  ISN 0038            DO 420 I = 1,NV
  ISN 0039            IA=VEH(1,I)
  ISN 0040            IB=VEH(2,I)
  ISN 0041            IC=VEH(3,I)
  ISN 0042            ID=VEH(4,I)
  ISN 0043            DO 286 J = K,KNM
  ISN 0044            L = J-K&1
  ISN 0045       *286 LZ(L)     = ITEM(VNM(1,J),I ,1)
  ISN 0046            WRITE(6,4100)I,STG(IA),STG(IB),STG(IC),STG(ID),(LZ(J), J=1,L)
  ISN 0047         420 CONTINUE
  ISN 0048            IF(NM.LE.KNM)  RETURN
  ISN 0050            IF(ITER.EQ.1)  WRITE(6,4001)
  ISN 0052            IF(ITER.EQ.2)  WRITE(6,4003)
  ISN 0054            IF(ITER.EQ.3)  WRITE (6,4004)
  ISN 0056         421 CONTINUE
  ISN 0057            RETURN
  ISN 0058        4000 FORMAT (1H1,34X,51HV E H I C L E / M I S S I O N   C A P A B I L I
                     1 T Y/46X,30H(1 = POSSIBLE, 0 = IMPOSSIBLE)/1HO,43X,10(2H1 ),
                     2  10(2H2 ),10(2H3 ),6(2H4 )/18H VEHICLE / MISSION,9X,4(20H1 2 3 4
                     35 6 7 8 9 0 ),9H1 2 3 4 5//)
  ISN 0059        4001 FORMAT(1H1/   27X  ,4(2H4 ),10(2H5 ),10(2H6 ),10(2H7 ),10(2H8 ),
                     1  2H9 /18H VEHICLE / MISSION,9X,9H6 7 8 9 0,4(20H 1 2 3 4 5 6 7 8
                     29 0)//)
  ISN 0060        4002 FORMAT (1HO,7X,14HMISSION NUMBER,   4X,45I2)
  ISN 0061        4003 FORMAT(1H1/   45X  ,36(2H1 )/ 27X,9(2H9 ),10(2H0 ),10(2H1 ),
                     1 10(2H2 ),6(2H3 )/
                     2   18H VEHICLE / MISSION, 9X,4(20H1 2 3 4 5 6 7 8 9 0 ),
                     3     9H1 2 3 4 5//)
  ISN 0062        4004 FORMAT (1H1/     27X,45(2H1 )/ 27X,4(2H3 ), 10(2H4 ), 10(2H5 ),
                     1  10(2H6 ), 10(2H7 ), 2H8 / 18H VEHICLE / MISSION, 9X,9H6 7 8 9 0,
                     2   4(20H 1 2 3 4 5 6 7 8 9 0)//)
  ISN 0063        4100 FORMAT (1H ,I2,1X,4(A4,1X), 2X,        45I2)
  ISN 0064            END

***** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02AL(R)
IEW0461  ITEM
IEW0461  PACK
IEW0461  IBCOM=

                                        CROSS REFERENCE TABLE

CONTROL SECTION                    ENTRY
  NAME     ORIGIN  LENGTH          NAME  LOCATION    NAME  LOCATION    NAME  LOCATION    NAME
AVAILI        00     7A8
SAVE1        7A8     FC4
SAV3        1770     980
SVACAV      20F0     B48
SAVALL      2C38    3A1C
TEMP        6658    4110

 LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION        LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
      358          SAVE1          SAVE1                   35C          SAV3           SAV3
      360          SVACAV         SVACAV                  364          SAVALL         SAVALL
      368          SAVALL         SAVALL                  36C          TEMP           TEMP
      370          ITEM           $UNRESOLVED             374          PACK           $UNRESOLVED
      378          IBCOM=         $UNRESOLVED             2C8          TEMP           TEMP
      2D0          SAVALL          SAVALL
ENTRY ADDRESS      00
TOTAL LENGTH     A768

****MOX02AL   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                              DATE  71.312/16.30.41

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
ISN 0002            SUBROUTINE CAPBLI
                C   VEHICLE DATA IS INPUT
                C  THE ORIGINAL CAPABILITY MATRIX BETWEEN VEHICLE AND MISSION IS SET UP
                C
ISN 0003             REAL ISP,LENT,NPERPD
ISN 0004             INTEGER*2 LTR,KODEV,NVS,MRV,NRP,NYP,KODEP,IVEHA,NTRIP,NPLS,NRR,
                    1  MR,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,MAPS,MAPF,MAPI,FINISH,
                    2  NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,VEH,NMULT,NONREC,NYD,
                    3  IS,MAT,LYR,LETT,LYD,MIN,MAF,MAIC
ISN 0005             COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                    1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                    2  RPLM(50),MAS(40,3), RXD(12,50)
ISN 0006             COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                    1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
ISN 0007             COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                    *             NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                    1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                    2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                    3   PSTGS(30,10,2)
ISN 0008             COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                    1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                    2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
ISN 0009             COMMON/SVACAV/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
                    1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
                    2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
ISN 0010             COMMON/SCRACH/IP,IV,IG,NPAX(2),NEH(4),NST(41),THRT(41),DIAM(41),
                    1  TSL(41),LENT(41),WTFU(41),WTIN(41),ISP(41),MZ(50),LZ(50),
                    2  NYPX(2),DUM(6369)
                C
ISN 0011             IF(IV.LT.0)  GO TO  14
ISN 0013             DO 2 I = 1,60
ISN 0014             ALPI(1,I) = .05
ISN 0015             ALPI(2,I) = .20
ISN 0016             ALPI(3,I) = .50
ISN 0017           2 ALPI(4,I) = .25
ISN 0018          14 DO 281 J = 1,61
ISN 0019             IF(IV.LT.0.AND.J.GT.KNV) RETURN
ISN 0021             IF(IV.LT.0.AND.IG.LT.0)  GO TO 27
ISN 0023             IF(IV.LT.0)  GO TO 15
ISN 0025             READ(5,106) (NEH(I), I=1,4),B1(J),B2(J),B3(J),B4(J),KOV
ISN 0026             IF(KOV.EQ.0)  GO TO 5002
ISN 0028             KODEV(J) = KOV
```

```
ISN 0029            DO 16 K = 1,4
ISN 0030          16 VEH(K,J) = NEH(K)
ISN 0031            READ(5,108) NSX,MVX,NPX, (NPAX(I), I=1,2),(NYPX(I), I=1,2), JKEY
ISN 0032            DO 17  I = 1,2
ISN 0033            NPAD(I,J) = NPAX(I)
ISN 0034          17 NYP(I,J) = NYPX(I)
ISN 0035            NVS(J) = NSX
ISN 0036            MRV(J) = MVX
ISN 0037            NRP(J) = NPX
ISN 0038            IF (JKEY.EQ.0) GO TO 15
ISN 0040            READ (5,114) (ALPI(I,J),I=1,4)
ISN 0041          15 DO 26 I = 1,4
ISN 0042            IF(VEH(I,J).EQ.0)  GO TO 27
ISN 0044            DO 25 K = 1,NSTG
ISN 0045            IF(VEH(I,J).NE.KODS(K))  GO TO 25
ISN 0047            VEH(I,J)  = K
ISN 0048            GO TO 26
ISN 0049          25 CONTINUE
ISN 0050          26 CONTINUE
ISN 0051          27 NV = J
ISN 0052            IF(IV.LT.0.AND.(IP.LT.0.OR.NP.EQ.0))  GO TO 9007
ISN 0054            DO 9008 I = 1,2
ISN 0055            IF (NPAD(I,J).EQ.0) GO TO 9008
ISN 0057            DO 9009 K = 1,NP
ISN 0058            IF (NPAD(I,J).NE.KODEP(K)) GO TO 9009
ISN 0060            NPAD(I,J) = K
ISN 0061            GO TO 9008
ISN 0062         9009 CONTINUE
ISN 0063         9008 CONTINUE
ISN 0064         9007 C1 = B1(J)
ISN 0065              C2 = B2(J)
ISN 0066              C3 = B3(J)
ISN 0067              C4 = B4(J)
ISN 0068              DO 28 I=1,NMIS
ISN 0069              NMULT(J,I) = 1
ISN 0070              LZ(I) = 1
ISN 0071              IF (IVEHA(I).EQ.0.OR.IVEHA(I).EQ.KODEV(J))  GO TO 21
ISN 0073              GO TO 8024
ISN 0074            21 VLX=VLR(I)-25573.
ISN 0075              IF(VLX.GE.C4-.01)  GO TO 8024
ISN 0077              WP=EXP(C1-C2*VLX-C3/(C4-VLX))
ISN 0078              IF(RPLM(I).LT.1.0.OR.IVEHA(I).NE.0)  GO TO 23

ISN 0080              DO 22 JJ = 1,4
ISN 0081              JJJ = 5-JJ
ISN 0082              IF(VEH(JJJ ,J).EQ.0)  GO TO 22
ISN 0084              LL = VEH(JJJ ,J)
ISN 0085              IF(RPLO(LL).LT..001)  GO TO 8024
ISN 0087              GO TO 23
ISN 0088            22 CONTINUE
ISN 0089            23 IF(WP.GE.WPR(I))    GO TO 24
ISN 0091              IF(WP.LE..001.OR.WPR(I)/WP.GE.100.)   GO TO 8024
ISN 0093              NMULT(J,I) = INT(WPR(I)/WP & .99)
ISN 0094              IF(NMULT(J,I).GT.NTRIP(I))  GO TO 8024
ISN 0096            24 IF (NOPT.NE.3.OR.IVEHA(I).NE.0)  GO TO 28
ISN 0098              IF(NPLS(I).EQ.0)  GO TO 8023
ISN 0100              IF(NPLS(I).NE.NVS(J))   GO TO 8024
ISN 0102         8023 IF (NRR(I).GT.NRP(J)) GO TO 8024
ISN 0104              IF (MR(I).EQ.0.OR.MRV(J).EQ.1)  GO TO 28
ISN 0106         8024 LZ(I) = 0
ISN 0107            28 CONTINUE
ISN 0108              CALL PACK(LZ,VM(1,J),NMIS,1)
ISN 0109           281 CONTINUE
ISN 0110              WRITE(6,113)
ISN 0111            99 RETURN
ISN 0112         5002 IF(NOPT.NE.2)  RETURN
ISN 0114              WRITE(6,111)
                C   CARDS MUST BE IN SAME ORDER AS INPUT STAGE CARDS
                C   ALL STAGES NOT TO BE USED IN MATCHING SCREEN MUST BE AT END OF DATA SET
ISN 0115              NTG = NSTG & 1
ISN 0116              DO 30 I = 1,NTG
ISN 0117              READ(5,109) J,NST(I),THRT(I),DIAM(I),TSL(I),LENT(I),WTFU(I),
                     1  WTIN(I),ISP(I)
ISN 0118              IF(J.EQ.0)  GO TO 31
ISN 0120              WRITE(6,112) I,NST(I),THRT(I),DIAM(I),TSL(I),LENT(I),WTFU(I),WTIN
                     1(I),ISP(I)
ISN 0121              IF(J.NE.KODS(I))   GO TO 5005
ISN 0123            30 CONTINUE
ISN 0124            31 CALL MATEI
ISN 0125         5004 RETURN
ISN 0126         5005 WRITE(6,110)
ISN 0127              RETURN
ISN 0128          106 FORMAT (4I2,4E13.6,18X,I2)
ISN 0129          108 FORMAT(3X,3I2,4I3,58X,I1)
ISN 0130          109 FORMAT(I4,I5,7F10.0)
```

```
ISN 0131        110 FORMAT(27HOSTAGE CARDS IN WRONG ORDER)
ISN 0132        111 FORMAT(1H1,8H STG NST,9X,4HTHRT,9X,4HDIAM,9X,4H TSL,9X,4HLENT,9X,
                   1   4HWTFU,9X,4HWTIN,10X,3HISP//)
ISN 0133        112 FORMAT(1HO,2I4,7F13.2)
ISN 0134        113 FORMAT (28HOMORE THAN 60 VEHICLES INPUT)
ISN 0135        114 FORMAT (3X,4F5.2)
ISN 0136          .   END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
           VARIABLE OPTIONS USED - SIZE=(126976,24576)                DEFAULT OPTION(S) USED
IEW0000      NAME MOX02CI(R)
IEW0461  PACK
IEW0461  MATEI
IEW0461  EXP
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH
CAPBLI	00	B40
SAVE1	B40	FC4
SAV3	1B08	980
SAV4	2488	3188
SAVALL	5610	3A1C
SVACAV	9030	B48
SCRACH	9878	6A60

ENTRY

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
220	SAVE1	SAVE1	224	SAV3	SAV3
228	SAV4	SAV4	22C	SAV4	SAV4
230	SAV4	SAV4	234	SAVALL	SAVALL
238	SAVALL	SAVALL	23C	SVACAV	SVACAV
240	SCRACH	SCRACH	244	PACK	$UNRESOLVED
248	MATEI	$UNRESOLVED	24C	EXP	$UNRESOLVED
250	IBCOM=	$UNRESOLVED	140	SCRACH	SCRACH
148	SAV3	SAV3			

ENTRY ADDRESS 00
TOTAL LENGTH 105D8

```
****MOX02CI   NOW REPLACED IN DATA SET
```

```
FORTRAN IV G LEVEL 1, MOD 4            CHOOZS            DATE = 71312          17/23/04

0001                SUBROUTINE CHOOZS
                C  DETERMINE OPTIMUM VEHICLE TO MISSION ASSIGNMENT
                C
0002                INTEGER*2  KOUT,LTR,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,
                   1  MINOPT,MORE,NSAVE,NADD,  NX,NINTYR,NTGYTR
0003                REAL NPERPD
                C
0004                COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                   1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0005                COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                   1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0006                COMMON/VARNCE/KSTAT,VARI(40),VARF(50),VARM(56),FMVAR(2,30),
                   1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)
0007                COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
                   1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
0008                COMMON/SCRACH/EXTRA,NADD,NX,MORE(10),ZKP,WKP,NXKP,LZKP(5),DUME(11)
                   *,  A2,LZ(46),W(500),W2(500),
                   1 TDS(500),WR(499),Z(500),COST(2,250),MINOPT(246,9),NODE(5,500),
                   2  NPOS,SIGSQ(9),ETC(9),
                   4  NCOST,LB,KX,KZ,NSAVE(10),KEEP(40),MZ(60),DUM
                C
                C
0009                IF(MYRS.GT.10) GO TO 2
0010                KI = 1
0011                KNEX = MYRS
0012                GO TO 7
0013              2 KI = 2
0014                KNEX = (MYRS + 1)/2
                C
                C       ***   INITIALIZE FUNCTIONS   ***
0015              7 NEXT=1
0016                NX=1
0017                KPNX = 10
0018                ZKP = 1.0E30
0019                NADD = 0
0020                NPOS = 0
0021                DO 16 I = 1,NUMD
0022             16 LZ(I) = 15
0023                CALL PACK(LZ,NODE(1,1),NUMD,4)
0024                DO 17 I = 1,8
0025             17 LZOPT(I) = 0
0026                DO 400 I = 1,10

0027            400 MORE(I) = 0
0028                IF(LP.GT.0)  WRITE (6,205)
                C
                C       ***   FIND W(1) = SUM OF COLUMN MINIMUMS OF FIRST CASE   ***
0029                W(1)=0.0
0030                W2(1) = 0.0
0031                TDS(1) = 0.0
0032                DO 19 J=1,NM
0033                COST(1,J) = 1.0E30
0034                COST(2,J) = 1.0E30
0035                IY = LYR(J)
0036                IF(IY.GT.MYRS)  GO TO 325
0037                JX = LETT(J)
0038                ITR = LTR(JX)
0039                CALL UNPACK(MZ,VNM(1,J),NV ,1)
0040                DO 18 I=1,NV
0041                IF(MZ(I).EQ.0)  GO TO 18
0042                X = NMULT(I ,JX)
0043                CX = YRLM(J)*RECUR(I ,IY,ITR)*X
0044                IF(CX.GE.COST(2,J))  GO TO 18
0045                IF(CX.LT.COST(1,J))  GO TO 176
0046                COST(2,J) = CX
0047                GO TO 18
0048            176 COST(2,J) = COST(1,J)
0049                COST(1,J)  = CX
0050                MIN(J) = I
0051             18 CONTINUE
0052                IF(COST(1,J).LT.1.0E25)  GO TO 20
0053            325 YRLM(J)=0.0
0054                MIN(J) = 0
0055                COST(1,J) = 0.0
0056                COST(2,J) = 0.0
0057             20 W2(1) = W2(1) + COST(2,J)
0058             19 W(1) = W(1) + COST(1,J)
0059                IF(NUMD.NE.0)  GO TO 25
0060                WRITE(6,211)  W(1)
0061            211 FORMAT(1H1//25H PROGRAM RECURRING COST =, F12.2)
0062                RETURN
                C
                C  PRESET SMALL SUST COSTS TO ZERO SO ALGORITHM IGNORES THEM IF MSOL.NE.1
                C   LOUT = NUMBER OF SUST COSTS GT 0 WHICH HAVE BEEN SET TO 0
                C
0063             25 IF(IFLAG.EQ.0)  LOUT = 0
```


30

```
0064              IF(MOS.EQ.0.OR.MOS.EQ.2)  GO TO 26
0065              CALL UNPACK(LZ,NODE(1,1),NUMD,4)
0066              DO 401 I = 1,NUMD
0067            401 KOUT(I) = I
0068              CALL POCSTI
0069              WRITE(6,211)  W(1)
                C
0070              CALL OUTPTI
                C
0071                .  RETURN
0072               26 IF(MSOL.EQ.1)  GO TO 29
0073              177 IF(NSOL.LE.1)  CALL UNPACK(LZ,NODE(1,1),NUMD,4)
0074              IF(IFLAG.GT.0)  GO TO 22
0075              DO 21 I = 1,NUMD
0076               21 KOUT(I) = 0
0077              X = MYRS
0078              G = 8.0/X
0079              IF(GUESS.LT.1.0E14) G = GUESS/(150.0*X)
0080              IF(SLO.GT.0.001)  G = SLO
0081              DO 27 I = 1,NUMD
0082              IF(SUST(I).LT..001)  GO TO 27
0083              IF(SUST(I).GT.G)  GO TO 27
0084              LOUT = LOUT + 1
0085              KOUT(I) = LOUT
0086              SAVS(LOUT) = SUST(I)
0087              SUST(I) = 0.0
0088              IF(DS(I).GE.1.)  GO TO 27
0089              IF(NSOL.LE.1)  LZ(I) = KNEX
0090              IF(NSOL.LE.1)  GO TO 27
0091              KOUT(I) = 0
0092              SUST(I) = SAVS(LOUT)
0093              LOUT = LOUT - 1
0094               27 CONTINUE
0095               28 IF(LOUT.GT.0.AND.NSOL.LE.1)  CALL PACK(LZ,NODE(1,1),NUMD,4)
0096                  GO TO 29
0097               22 IF(LOUT.EQ.0.OR.NSOL.GT.1)  GO TO 29
0098                  DO 23 I = 1,NUMD
0099                  IF(KOUT(I).EQ.0.OR.DS(I).GE.1.)  GO TO 23
0100                  LZ(I) = KNEX
0101               23 CONTINUE
0102                  CALL PACK(LZ,NODE(1,1),NUMD,4)
                C
                C     ***   PICK COST TO CONSIDER NEXT   ***
0103               29 NCOST = 0
0104                  NKEY = 0
0105                  FMAX = -1.0E35
0106                  IF (KPNX.NE.NX)
                     1CALL UNPACK (LZ,NODE(1,NX),NUMD,4)
0107               30 DO 35 NIC = 1,NUMD
0108                  IF(LZ(NIC).LT.15) GO TO 35
0109                  NKEY = NKEY + 1
0110                  IF(KPNX.EQ.NX)  GO TO 300
0111                  WR(NIC) = 0.0
0112                  DO 33 J= 1,NM
0113                  IF(YRLM(J).LT..001)  GO TO 33
0114                  CALL UNPACK(MZ,VNM(1,J),NV ,1)
0115                  CMIN = 1.0E30
0116                  KO = LYR(J)
0117                  JX = LETT(J)
0118                  ITR = LTR(JX)
0119                  DO 32 I1 = 1,NV
0120                  IF(MZ(I1).EQ.0)   GO TO 32
0121                  I = I1
0122                  IF(ITR.EQ.2)  I = I1 + NV
0123                  DO 31 M = 1, 20
0124                  IF(NONREC(I,M).EQ.0)  GO TO 315
0125                  NO = NONREC(I,M)
0126                  IF(NO.EQ.NIC)  GO TO 32
0127                  IF(K1*LZ(NO).LT.KO)  GO TO 32
0128               31 CONTINUE
0129              315 X = NMULT(I1,JX)
0130                  CX = YRLM(J)*RECUR(I1,KO,ITR)*X
0131                  IF(CX.LT.CMIN)  CMIN = CX
0132               32 CONTINUE
0133                  WR(NIC) = WR(NIC) + CMIN
0134               33 CONTINUE
0135              300 PF = WP(NIC) - W(NX)
0136                  IF(PF.LT..001)   GO TO 35
0137              301 IF(SUST(NIC).GE..001) DF = DS(NIC)*0.5 + SUST(NIC) + PF
                     1  -1.0E4/(SUST(NIC)**4)
0138                  IF(SUST(NIC).LT..001)  DF = 0.5*DS(NIC) + 4.0 + PF
0139                  IF(SUST(NIC).LT..001.AND.PF.GT.1.0E10)   DF = 1.0E34
0140                  IF(DF.LE.FMAX)  GO TO 35
0141                  FMAX = DF
0142                  NCOST = NIC
0143               35 CONTINUE
```

```
0144          36 IF(NCOST.EQ.0)  GO TO 73
                C
                C     ALLOCATE SPACE FOR NEW NODES
                C
0145         295 IF(SUST(NCOST).GE..001)  J=0 + (NYD(NCOST) - 1)/KI
0146             IF(SUST(NCOST).LT..001)   J = KNEX - 1
0147             IF(NEXT.EQ.1)  GO TO 41
0148             DO 40  I = 2,NEXT
0149             K = NEXT + 2 - I
0150             IF(Z(K).LE.GUESS)  GO TO 40
0151             J=J+1
0152             IF(SUST(NCOST).GE..001)  NSAVE(J) = K
0153             IF(SUST(NCOST).LT..001)  NSAVE(1) = K
0154             IF(J.EQ.KNEX) GO TO 44
0155          40 CONTINUE
0156          41 IF(J.EQ.KNEX)  GO TO 44
0157             J=J+1
0158             NEXT=NEXT+1
0159             IF(NEXT.EQ.500) GO TO 74
0160             IF(SUST(NCOST).GE..001)  NSAVE(J) = NEXT
0161             IF(SUST(NCOST).LT..001)  NSAVE(1) = NEXT
0162             GO TO 41
                C
                C     ***   BRANCH WITH VARYING YEARS OF SUSTAINING COST   ***
0163          44 DO 52 K=1,10
0164             IF(SUST(NCOST).GE..001.AND.K.LT.1+(NYD(NCOST)-1)/KI)  GO TO 52
0165             KX=NSAVE(K)
0166             IF((K-1)*KI.LT.LYD(NCOST)) GO TO 45
0167             W(KX) = 1.0E30
0168             Z(KX) =20.0E30
0169             GO TO 509
0170          45 DO 46 I=1,5
0171          46 NODE(I,KX)=NODE(I,NX)
0172             LZ(NCOST)=K-1
0173             IF(K.EQ.1 + (NYD(NCOST)-1)/KI)   LZ(NCOST) = 0
0174             CALL PACK(LZ,NODE(1,KX),NUMD,4)
0175             LB = K-1
                C
0176             CALL LBONDI
                C
0177         509 IF(SUST(NCOST).LT..001) GO TO 53
0178             IF(KI*K.GE.MYRS) GO TO 53
0179          52 CONTINUE



                C
                C     ***   BRANCH INCLUDING NCOST AND ALL SUSTAINING - PUT IN NODE NX   ***
0180          53 LZ(NCOST) = (LYD(NCOST) + KI - 1)/KI
0181             CALL PACK (LZ,NODE(1,NX),NUMD,4)
0182             IF(W(NX).GT.W(KX)-.0001.AND.W2(NX).GT.W2(KX)-0.0001)  LB= 50
0183             IF(W(NX).GT.W(KX)-.0001.AND.W2(NX).GT.1.0E25.AND.W2(KX)-W2(NX).LT.
                1  1.0E25)  LB= 50
0184             KX = NX
0185             KZ = LYD(NCOST)
                C
0186             CALL LBONDI
                C
                C
                C     PICK NEXT NODE FOR BRANCHING AS THE ONE WITH LEAST LOWER BOUND Z
0187          55 KPNX = NX
0188             NX = 1
0189             DO 59  I=2,NEXT
0190             IF(Z(NX).GT.Z(I))  NX = I
0191          59 CONTINUE
0192             IF(Z(NX).LE.GUESS)  GO TO 29
0193             IF(NADD.GT.0)  GO TO 60
0194             WRITE(6,202)
0195             GUESS = 0.0
0196             RETURN
0197          60 WRITE(6,206) Z(NX)
0198             GO TO 109
0199          74 WRITE(6,203)
0200             IF(NADD.GT.0)  GO TO 109
0201             RETURN
                C
0202          73 IF(NKEY.EQ.0)  GO TO 75
0203             DO 37 I = 1,NUMD
0204             IF(LZ(I).EQ.15)  LZ(I) = 0
0205          37 CONTINUE
0206             CALL PACK(LZ,NODE(1,NX),NUMD,4)
                C
                C              ASSIGN VEHICLE TO MISSION
0207          75 DO 80 J=1,NM
0208             IF(YRLM(J).LT.0.0001)  GO TO 79
0209             CALL UNPACK(MZ,VNM(1,J),NV ,1)
0210             CMIN=1.0E30
0211             KO = LYR(J)
0212             JX = LETT(J)
```

```
0213              ITR = LTR(JX)
0214              DO 78 I1 = 1,NV
0215              IF(MZ(I1).EQ.0)  GO TO 78
0216              I = I1
0217              IF(ITR.EQ.2)  I = I1 + NV
0218              DO 77 K=1,20
0219              IF(NONREC(I,K).EQ.0) GO TO 775
0220              NO =NONREC(I,K)
0221              IF(KI*LZ(NO).LT.KO)      GO TO 78
0222           77 CONTINUE
0223          775 X = NMULT(I1,JX)
0224              CX=YRLM(J)*RECUR(I1,KO,ITR)*X
0225              IF(CX.GE.CMIN) GO TO 78
0226              CMIN=CX
0227              MIN(J)=I1
0228           78 CONTINUE
0229              GO TO 80
0230           79 MIN(J) = 0
0231           80 CONTINUE
0232              IF(NPOS.EQ.0)   GO TO 85
0233              DO 355 I = 1,10
0234              IF(MORE(I).EQ.0)  GO TO 356
0235              IF(MORE(I).EQ.NX)   GO TO 354
0236          355 CONTINUE
0237          356 DO 82 NA = 1,NPOS
0238              DO 81 J = 1,NM
0239              IF(MIN(J).NE.MINOPT(J,NA))  GO TO 82
0240           81 CONTINUE
0241              IF(LP.GT.0)
                 1WRITE(6,204) NX,Z(NX), NA
0242              GO TO 103
0243           82 CONTINUE
0244           85 IF(NP.EQ.0.AND.LOUT.EQ.0)  GO TO 86
                C
0245          354 CALL PDCSTI
                C
0246              IF(GUESS.LT..001)  GO TO 109
0247              IF(Z(NX).GT.19.0E30)  GO TO 55
0248              IF(EXTRA.LT.1.0)  GO TO 86
0249              IF(NPOS.EQ.9)  GO TO 84
0250              GUESS = 2.0*Z(NX)
0251              NPOS = 1 + NPOS
0252              IF(ZKP.LE.Z(NX))  GO TO 318

0253              ZKP = Z(NX)
0254              WKP = W(NX)
0255              NXKP = NX
0256              KPNPOS = NPOS
0257              DO 317 I = 1,5
0258           317 LZKP(I) = NODE(I,NX)
0259           318 DO 319 I = 1,NM
0260           319 MINOPT(I,NPOS) = MIN(I)
0261              GO TO 55
0262           84 IF(NADD.GT.0)  GO TO 109
0263              NX = NXKP
0264              Z(NX) = ZKP
0265              W(NX) = WKP
0266              CALL UNPACK(LZ,LZKP(1),NUMD,4)
0267              DO 87 I = 1,NM
0268           87 MIN(I) = MINOPT(I,KPNPOS)
0269              GO TO 354
0270           86 NADD = NADD + 1
0271              NBDD = NADD
0272              NPOS = MAXO(NBDD,NPOS)
0273              DMIN = Z(NX) - W(NX)
0274              WRITE(6,201) NADD,NX,W(NX), DMIN, Z(NX)
                C
0275              CALL OUTPTI
                C
0276              ETC(NADD) = Z(NX)
0277              IF(KSTAT.GT.0)  CALL CMPARE
0278              IF((IFLAG.EQ.0.AND.LCK.EQ.1).OR.(NPOS.GE.10))  RETURN
0279              IF(NADD.LT.NSOL)  GO TO 101
0280              IF(NADD.EQ.1)  RETURN
0281          109 DO 110 I = 1,NM
0282          110 MIN(I) = MINOPT(I,1)
0283              RETURN
                C
                C   STORE OPTIMAL VALUES
0284          101 DO 102 I = 1,NM
0285          102 MINOPT(I,NADD) = MIN(I)
0286          103 Z(NX) = 1.0E30
0287              NX = 1
0288              GO TO 55
0289           201 FORMAT (1H1,13(1H*),32H S O L U T I O N   N U M B E R  ,I2,12(1H*)
                  1  /1H ,13,4X,11HRECURRING =, F10.2,3X,14HNONRECURRING =,F10.2,3X,
                  2  27HTOTAL LAUNCH VEHICLE COST =, F10.2)
```

```
FORTRAN IV G LEVEL 1, MOD 4            CHOOZS            DATE = 71312          17/23/04

 0290          202 FORMAT(14H1GUESS TOO LOW)
 0291          203 FORMAT (19H1EXCEEDED 500 NODES)
 0292          204 FORMAT (23HOTHE ASSIGNMENT AT NODE,I4,3X,16HWITH LOWER BOUND,F10.2
                  1,  3X,35HIS THE SAME AS SOLUTION/POSSIBILITY, I4)
 0293          205 FORMAT (1H1,55HB R A N C H   A N D   B O U N D   N O D E   V A L U
                  1 E S/58HONODE  BRANCHED  COST   YEARS    RECURRING   NON-RECURRING
                  2,5X,5HTOTAL/13H  NO.    FROM,5X,12HNO.  SUSTAIN,5X,3(5HBOUND,9X)/)
 0294          206 FORMAT(28HONEXT SOLUTION HAS VALUE GT ,F10.2)
 0295              END


TOTAL MEMORY REQUIREMENTS 002280 BYTES


 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
           VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOXO2CH(R)
IEW0461  PACK
IEW0461  IBCOM=
IEW0461  UNPACK
IEW0461  PDCSTI
IEW0461  OUTPTI
IEW0461  LBONDI
IEW0461  CMPARE
IEW0461  MAXO

                                                  MODULE MAP

  CONTROL SECTION                          ENTRY
    NAME     ORIGIN  LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  CHOOZS         00    2280
  SAV3         2280     980
  SAVALL       2C00    3A1C
  VARNCE       6620     ADC
  TEMP         7100    4110
  SCRACH       B210    6A60

 ENTRY ADDRESS       00
 TOTAL LENGTH     11C70

****MOXO2CH   NOW REPLACED IN DATA SET
```

```
FORTRAN IV G LEVEL 1, MOD 4            CMPARE            DATE = 71312          17/07/40

0001                SUBROUTINE CMPARE
                C  STATISTICALLY COMPARE ASSIGNMENTS FOUND
                C
0002                INTEGER*2NU,NBY,NOB,     MODE,FINISH,NSTG,LABF,LABS,LABI,NFML,NFMU,
                   1  KODS,MAS,LTR,MAF,MAIC,NPAD,NPFAM,NPINTL,NPINTU,NFS,NPSTG,MAPS,
                   2  MAPF,MAPI,NYD,MAT,LYD,IS,LYR,LETT,MIN,VEH,NONREC,NMULT,KOUT,
                   3  NINTYR,NTGYTR,MINOPT,MORE,NADD,NX,NRCU
0003                COMMON/SAVSAR/COR,POJ(3),SRJ(3,3),NU(40),NBY(40),NOB(40),RINT(40),
                   1    PLCINT(40),XLT(40), PLCT(40),UPP(40),TAT(40),TAMT(50),SR(40,3),
                   2    MODE(40,3),PLC(40,3)
0004                COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                   1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                   2  RPLM(50),MAS(40,3), RXD(12,50)
0005                COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MDS,NMIS,NSPR,NPERPD(30),
                   1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0006                COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                   *              NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                   1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                   2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                   3    PSTGS(30,10,2)
0007                COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                   1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0008                COMMON/VARNCE/KSTAT,VART(40),VARF(50),VARM(56),FMVAR(2,30),
                   1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)
0009                COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
                   1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
0010                COMMON/SCRACH/EXTRA,NADD,NX,MORE(10),ZKP,WKP,NXKP,LZKP(5),DUME(11)
                   *,   A2,LZ(46),W(500),W2(500),
                   1 TDS(500),WR(499),Z(500),COST(2,250),MINOPT(246,9),NODE(5,500),
                   2  NPOS,SIGSQ(9),ETC(9),
                   3  TSTG(40,2),NRCU(40),DUM(10)
0011                DIMENSION TRINT(40)
                C
0012                DO 50 I = 1,NSTG
0013                NRCU(I) = 0
0014                TSTG(I,1) = 0.0
0015              50 TSTG(I,2) = 0.0
0016                IF(NCI.EQ.0)  GO TO 70
0017                DO 60 I = 1,NCI
0018              60 TRINT(I) = 0.0
                C
                C    CALCULATE NUMBER OF TIMES EACH RECURRING COST IS USED

0019              70 DO 100 J = 1,NM
0020                IF(YRLM(J).LT..001)  GO TO 100
0021                I = MIN(J)
0022                JX = LETT(J)
0023                ITR = LTR(JX)
0024                X = NMULT(I,JX)
0025                DO 99 MS = 1,4
0026                L = VEH(MS,I)
0027                IF(L.EQ.0)  GO TO 100
0028                TSTG(L,ITR) = TSTG(L,ITR) + YRLM(J)*X
0029                IF(NCI.EQ.0)  GO TO 99
0030                IF(MS.EQ.4)  GO TO 100
0031                IF(VEH(MS+1,I).EQ.0)  GO TO 100
0032                L1 = VEH(MS+1,I)
0033                DO 98 MI = 1,NCI
0034                DO 96 KY = 1,4
0035                IF(NFML(MI).NE.NFS(L,KY))   GO TO 96
0036                DO 95 KZ = 1,4
0037                IF(NFMU(MI).EQ.NFS(L1,KZ))  GO TO 97
0038              95 CONTINUE
0039              96 CONTINUE
0040                GO TO 98
0041              97 TRINT(MI) = TRINT(MI) + YRLM(J)*X
0042              98 CONTINUE
0043              99 CONTINUE
0044             100 CONTINUE
0045                TOT = 0.0
0046                VTC = 0.0
0047                ATC = 0.0
                C
                C    CALCULATE VARIANCES AND CORRELATE DEV. COSTS TO OPERATING COSTS
0048             400 DO 500 I = 1,NUMD
0049                IF(LZ(I).EQ.0)  GO TO 500
0050                IF(NADD.GT.1)  LZ(I) = LZ(I)*KI
0051                XX = LZ(I) - NYD(I) + 1
0052                SU = SUST(I)
0053                IF(KOUT(I).EQ.0)  GO TO 402
0054                LT = KOUT(I)
0055                SU = SAVS(LT)
0056             402 TOT = TOT + DS(I) + XX*SU
0057                J = MAT(I)
0058                IF(J.GT.1000)  J = J - 2000
0059                IF(J.LT.-200)  GO TO 499
```

```
0060              IF(J.LT.-100)  GO TO 440
0061              IF(J.LT.0)  GO TO 470
0062              IF(MAS(J,1).NE.I)  GO TO 499
0063              TSR = 0.0
0064              DTM = 0.0
                C  TOP = TOTAL OPERATING;   TSIG = TOTAL OPERATING VARIANCE
0065              TOP = 0.0
0066              TSIG = 0.0
0067              IF(LABS(J).EQ.0)  GO TO 410
0068              L = LABS(J)
0069              DO 405 K = 1,12
0070            405 DTM = DTM + RXD(K,L)
0071              TSR = (EXP(1.5*VARF(L))*DTM)/(1.0 + EXP(1.5*VARF(L)))
0072              ATC = ATC + TSR
0073              IF(VARF(L).LT..001)  GO TO 410
0074              TTSR = TSR*TSR*(EXP(VARF(L)) - 1.0)
0075              VTC = VTC + TTSR
0076            410 DTM = DS(I) - DTM
0077              DXM = TSR
0078              IF(NU(J).EQ.0)  GO TO 420
0079              X = NU(J)
0080              X = ABS(X)
0081              TSR = (EXP(1.5*VARI(J))*X*UPP(J))/(1.0 + EXP(1.5*VARI(J)))
0082              ATC = ATC + TSR
0083              IF(VARI(J).LT..001)   GO TO 411
0084              TTSR = TSR*TSR*(EXP(VARI(J)) - 1.0)
0085              VTC = VTC + TTSR
0086              TSIG = VARI(J)
0087            411 TOP = TSR
0088              DTM = DTM - X*UPP(J)
0089            420 IF(SVAR(4,J).LT..001)  ATC = ATC + .5*DTM
0090              IF(SVAR(4,J).LT..001)  GO TO 421
0091              TSR = (EXP(1.5*SVAR(4,J))*DTM)/(1.0 + EXP(1.5*SVAR(4,J)))
0092              ATC = ATC + .5*TSR
0093              TTSR = TSR*TSR*(EXP(SVAR(4,J)) - 1.0)
0094              VTC = VTC + TTSR
0095              DXM = DXM + TSR
0096            421 IF(SVAR(5,J).LT..001)  ATC = ATC + .5*XX*SU
0097              IF(SVAR(5,J).LT..001.OR.SU     .LT..001)  GO TO 422
0098              TSR = (EXP(1.5*SVAR(5,J))*XX*SU)/(1.0 + EXP(1.5*SVAR(5,J)))
0099              ATC = ATC + TSR
0100              TTSR = TSR*TSR*(EXP(SVAR(5,J)) - 1.0)
0101              VTC = VTC + TTSR
0102              TOP = TOP + TSR
0103              IF(SVAR(5,J).GT.TSIG)  TSIG = SVAR(5,J)
                C  ADD STAGE RECURRING AND ASSOCIATED VARIANCES
0104            422 NRCU(J) = 1
0105              TSR = SR(J,1)*(TSTG(J,1) + TSTG(J,2))
0106              TOT = TOT + TSR
0107              IF(SVAR(1,J).LT..001)  ATC = ATC + .5*TSR
0108              IF(TSR.LT..001.OR.SVAR(1,J).LT..001)  GO TO 423
0109              TSR = (EXP(1.5*SVAR(1,J))*TSR)/(1.0 + EXP(1.5*SVAR(1,J)))
0110              ATC = ATC + TSR
0111              TOP = TOP + TSR
0112              IF(SVAR(1,J).GT.TSIG)  TSIG = SVAR(1,J)
0113              TTSR = TSR*TSR*(EXP(SVAR(1,J)) - 1.0)
0114              VTC = VTC + TTSR
0115            423 TSR = SR(J,2)*TSTG(J,1)
0116              TOT = TOT + TSR
0117              IF(SVAR(2,J).LT..001)  ATC = ATC + .5*TSR
0118              IF(TSR.LT..001.OR.SVAR(2,J).LT..001)  GO TO 424
0119              TSR = (EXP(1.5*SVAR(2,J))*TSR)/(1.0 + EXP(1.5*SVAR(2,J)))
0120              ATC = ATC + TSR
0121              TOP = TOP + TSR
0122              IF(SVAR(2,J).GT.TSIG)  TSIG = SVAR(2,J)
0123              TTSR = TSR*TSR*(EXP(SVAR(2,J)) - 1.0)
0124              VTC = VTC + TTSR
0125            424 TSR = SR(J,3)*TSTG(J,2)
0126              TOT = TOT + TSR
0127              IF(SVAR(3,J).LT..001)  ATC = ATC + .5*TSR
0128              IF(TSR.LT..001.OR.SVAR(3,J).LT..001)  GO TO 425
0129              TSR = (EXP(1.5*SVAR(3,J))*TSR)/(1.0 + EXP(1.5*SVAR(3,J)))
0130              ATC = ATC + TSR
0131              TOP = TOP + TSR
0132              IF(SVAR(3,J).GT.TSIG)  TSIG = SVAR(3,J)
0133              TTSR = TSR * TSR*(EXP(SVAR(3,J)) - 1.0)
0134              VTC = VTC + TTSR
0135            425 IF(COR.LE..001.OR.SVAR(4,J).LT..001.OR.TSIG.LT..001)  GO TO 500
0136              S1 = SQRT(SVAR(4,J))
0137              S2 = SQRT(TSIG)
0138              TTSR = DXM*TOP*(EXP(COR*S1*S2) - 1.0)
0139              VTC = VTC + 2.0*TTSR
0140              GO TO 500
0141            440 JX = -J -100
0142              IF(MAIC(JX,1).NE.I)  GO TO 499
0143              DTM = 0.0
```

```
0144              IF(LABI(JX).EQ.0)  GO TO 450
0145              L = LABI(JX)
0146              DO 445  K = 1,12
0147            445 DTM = DTM + RXD(K,L)
0148              IF(VARF(L).LT..001)  ATC = ATC + .5*DTM
0149              IF(VARF(L).LT..001)  GO TO 450
0150              TSR = (EXP(1.5*VARF(L))*DTM)/(1.0 + EXP(1.5*VARF(L)))
0151              ATC = ATC + TSR
0152              TTSR = TSR*TSR*(EXP(VARF(L)) - 1.0)
0153              VTC = VTC + TTSR
0154            450 DTM = DS(I) - DTM
0155              IF(FIVAR(2,JX).LT..001)  ATC = ATC + .5*DTM
0156              IF(FIVAR(2,JX).LT..001)  GO TO 451
0157              DTM = (EXP(1.5*FIVAR(2,JX))*DTM)/(1.0 + EXP(1.5*FIVAR(2,JX)))
0158              ATC = ATC + DTM
0159              TTSR = DTM*DTM*(EXP(FIVAR(2,JX)) - 1.0)
0160              VTC = VTC + TTSR
0161            451 IF(FIVAR(3,JX).LT..001)  ATC = ATC + .5*SU*XX
0162              IF(SU     .LT..001.OR.FIVAR(3,JX).LT..001)  GO TO 500
0163              TSR = (EXP(1.5*FIVAR(3,JX))*SU*XX)/(1.0 + EXP(1.5*FIVAR(3,JX)))
0164              ATC = ATC + TSR
0165              TTSR = TSR*TSR*(EXP(FIVAR(3,JX)) - 1.0)
0166              VTC = VTC + TTSR
0167            452 IF(COR.LT..001.OR.FIVAR(2,JX).LT..001)  GO TO 500
                C  THERE IS NO CORRELATION BETWEEN INTEGRATION RECURRING AND DEVELOPMENT COSTS
0168              S2 = SQRT(FIVAR(2,JX))
0169              S1 = SQRT(FIVAR(3,JX))
0170              TTSR = DTM*TSR*(EXP(COR*S1*S2) - 1.0)
0171              VTC = VTC + TTSR
0172              GO TO 500
0173            470 JX = -J
0174              IF(MAF(JX,1).NE.I)  GO TO 499
0175              DTM = 0.0
0176              IF(LABF(JX).EQ.0)  GO TO 480
0177              L = LABF(JX)
0178              DO 475 K = 1,12
0179            475 DTM = DTM + RXD(K,L)
0180              IF(VARF(L).LT..001)  ATC = ATC + .5*DTM
0181              IF(VARF(L).LT..001)  GO TO 480
0182              TSR = (EXP(1.5*VARF(L))*DTM)/(1.0 + EXP(1.5*VARF(L)))
0183              ATC = ATC + TSR
0184              TTSR = TSR*TSR*(EXP(VARF(L)) - 1.0)
0185              VTC = VTC + TTSR

0186            480 DTM = DS(I) - DTM
0187              IF(FMVAR(1,JX).LT..001)  ATC = ATC + .5*DTM
0188              IF(FMVAR(1,JX).LT..001)  GO TO 481
0189              DTM = (EXP(1.5*FMVAR(1,JX))*DTM)/(1.0 + EXP(1.5*FMVAR(1,JX)))
0190              ATC = ATC + DTM
0191              TTSR = DTM*DTM*(EXP(FMVAR(1,JX)) - 1.0)
0192              VTC = VTC + TTSR
0193            481 IF(FMVAR(2,JX).LT..001)  ATC = ATC + .5*SU*XX
0194              IF(SU     .LT..001.OR.FMVAR(2,JX).LT..001)  GO TO 500
0195              TSR = (EXP(1.5*FMVAR(2,JX))*SU*XX)/(1.0 + EXP(1.5*FMVAR(2,JX)))
0196              ATC = ATC + TSR
0197              TTSR = TSR*TSR*(EXP(FMVAR(2,JX)) - 1.0)
0198              VTC = VTC + TTSR
0199            482 IF(COR.LT..001.OR.FMVAR(1,JX).LT..001)  GO TO 500
0200              S1 = SQRT(FMVAR(1,JX))
0201              S2 = SQRT(FMVAR(2,JX))
0202              TTSR = DTM*TSR*(EXP(COR*S1*S2) - 1.0)
0203              VTC = VTC + TTSR
0204              GO TO 500
0205            499 ATC = ATC + .5*(DS(I) + XX*SU)
0206            500 CONTINUE
                C
                C   CALCULATE VARIANCE DUE TO RECURRING COSTS WHICH HAVE NOT BEEN CONSIDERED YET
0207              DO 200 L = 1,NSTG
0208              IF(NRCU(L).EQ.1)  GO TO 200
0209              TSR = SR(L,1)*(TSTG(L,1)+TSTG(L,2))
0210              TOT = TOT + TSR
0211              IF(SVAR(1,L).LT..001)  ATC = ATC + .5*TSR
0212              IF(TSR.LT..001.OR.SVAR(1,L).LT..001)  GO TO 110
0213              TSR = (EXP(1.5*SVAR(1,L))*TSR)/(1.0 + EXP(1.5*SVAR(1,L)))
0214              ATC = ATC + TSR
0215              TTSR = TSR*TSR*(EXP(SVAR(1,L)) - 1.0)
0216              VTC = VTC + TTSR
0217            110 TSR = SR(L,2)*TSTG(L,1)
0218              TOT = TOT + TSR
0219              IF(SVAR(2,L).LT..001)  ATC = ATC + .5*TSR
0220              IF(TSR.LT..001.OR.SVAR(2,L).LT..001)  GO TO 111
0221              TSR = (EXP(1.5*SVAR(2,L))*TSR)/(1.0 + EXP(1.5*SVAR(2,L)))
0222              ATC = ATC + TSR
0223              TTSR = TSR*TSR*(EXP(SVAR(2,L)) - 1.0)
0224              VTC = VTC + TTSR
0225            111 TSR = SR(L,3)*TSTG(L,2)
0226              TOT = TOT + TSR
```

```
FORTRAN IV G LEVEL 1, MOD 4            CMPARE            DATE = 71312          17/07/40

0227              IF(SVAR(3,L).LT..001)  ATC = ATC + .5*TSR
0228              IF(TSR.LT..001.OR.SVAR(3,L).LT..001)  GO TO 200
0229              TSR = (EXP(1.5*SVAR(3,L))*TSR)/(1.0 + EXP(1.5*SVAR(3,L)))
0230              ATC = ATC + TSR
0231              TTSR = TSR * TSR*(EXP(SVAR(3,L)) - 1.0)
0232              VTC = VTC + TTSR
0233            200 CONTINUE
0234              IF(NCI.EQ.0)  GO TO 510
0235              DO 300 I = 1,NCI
0236             *  TSR = RINT(I)*TRINT(I)
0237              TOT = TOT + TSR
0238              IF(FIVAR(1,I).LT..001)  ATC = ATC + .5*TSR
0239              IF(TSR.LT..001.OR.FIVAR(1,I).LT..001)  GO TO 300
0240              TSR = (EXP(1.5*FIVAR(1,I))*TSR)/(1.0 + EXP(1.5*FIVAR(1,I)))
0241              ATC = ATC + TSR
0242              TTSR = TSR*TSR*(EXP(FIVAR(1,I)) - 1.0)
0243              VTC = VTC + TTSR
0244            300 CONTINUE
                C
0245            510 IF(NADD.LE.1)  A2 = ATC
0246              IF(NADD.GT.1)  ATC = ETC(NADD) - ETC(1) + A2
0247              SIGSQ(NADD) = ALOG(ATC*ATC + VTC) - ALOG(ATC*ATC)
0248              XMODE = ATC*(EXP(-1.5*SIGSQ(NADD)))
0249              TP = SQRT(SIGSQ(NADD))
0250              XMU = ALOG(ATC) - .5*SIGSQ(NADD)
0251              VTC = SQRT(VTC)
0252              XX = XMODE + ETC(NADD) - ATC
0253              WRITE(6,900) NADD,ETC(NADD),TOT,XX,   VTC,XMU,SIGSQ(NADD)
0254            900 FORMAT ('1SOLUTION',I3,'  HAS EXPECTED L V COST', F10.2,'  (',
                   *  F10.2,')  ', 3X,'MODE =',F10.2,3X,'STD. DEV. =',F10.2//
                   *  ' PARAMETERS MU AND SIGMASQ =',F10.2,3X,'AND',F10.2)
0255              XMD = (ALOG(XMODE) - XMU)/TP
0256              CALL NDTR(XMD,P2,D)
0257              XDUM = (ATC/TP)*(1.00/2.5066)
0258              C = (XDUM/XMODE)*EXP(-.5*XMD**2)
0259              WRITE(6,910) XX,P2,C
0260            910 FORMAT (1HO, ' PROB (COST LE', F10.0,' )  =', F4.2,3X,'DENSITY =',
                   2  F10.4)
0261              P2 = P2 + .5
0262              CALL NDTRI(P2,Y2,C,IE)
0263              Z2 = EXP(TP*Y2 + XMU)
0264              C = (XDUM/Z2)*EXP(-.5*Y2**2)
0265              Z2 = Z2 + ETC(NADD) - ATC

0266              WRITE(6,905) Z2,P2,XX,   Z2,C
0267            905 FORMAT('0PROB (COST LE', F10.0,' )  =',F4.2,3X,
                   *' 50 PERCENT UNCERTAINTY INTERVAL =',F10.0,2X,'TO', F10.0,3X,
                   *  'DENSITY =',F10.2)
0268              XX = ETC(NADD) - ATC
0269              WRITE(6,911) XX
0270            911 FORMAT(1HO/ 1HO, ' PROB (COST LE', F10.0,' )  =', ' .00', 3X,
                   *   'DENSITY =      .00')
0271              P = .1
0272              DO 520 I = 1,5
0273              CALL NDTRI(P,Y,C,IE)
0274              X5 = EXP(TP*Y + XMU)
0275              D = (XDUM/X5)*EXP(-.5*Y**2)
0276              X5 = X5 + ETC(NADD) - ATC
0277              WRITE(6,910) X5,P,D
0278            520 P = P + .2
0279              IF(NADD.LE.1)  GO TO 700
0280              NT = NADD - 1
0281              DO 600 I = 1,NT
0282              RHO = -.3
0283              TP1 = SQRT(SIGSQ(I))
0284              ATC = ETC(I) - ETC(1) + A2
0285              XMU1 = ALOG(ATC) - .5*SIGSQ(I)
0286              DO 590 J = 1,4
0287              RHO = RHO + .3
0288              Y =(XMU - XMU1)/(SQRT(SIGSQ(I) + SIGSQ(NADD) - 2.0*RHO*TP*TP1))
0289              CALL NDTR(Y,P,D)
0290              WRITE(6,901) NADD,I,P,RHO
0291            901 FORMAT( '0 PROB ( ASSIGNMENT',I3,' COST GE ASSIGNMENT',I3,' COST)
                   *=', F4.2,'  IF CORRELATION =', F3.1//)
0292            590 CONTINUE
0293            600 CONTINUE
0294            700 WRITE(6,906)
0295            906 FORMAT(1H1)
0296              RETURN
0297              END
```

```
TOTAL MEMORY REQUIREMENTS 002870 BYTES


F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000       NAME MOX02CM(R)
IEW0461   IBCOM=
IEW0461   NDTR
IEW0461   NDTR1
IEW0461   EXP
IEW0461   SQRT
IEW0461   ALOG

                                                        MODULE MAP

  CONTROL SECTION                              ENTRY
    NAME     ORIGIN   LENGTH                    NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  CMPARE         00     2870
  SAVSAR       2870      A5C
  SAVE1        35D0      FC4
  SAV3         4598      980
  SAV4         4F18     3188
  SAVALL       80A0     3A1C
  VARNCE       BAC0      ADC
  TEMP         C5A0     4110
  SCRACH       106B0     6A60

 ENTRY ADDRESS        00
 TOTAL LENGTH      17110

****MOX02CM   NOW REPLACED IN DATA SET


(17)          OS/360  FORTRAN H                                DATE  71.312/17.03.19

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
 ISN 0002             SUBROUTINE CONSTR
               C     DETERMINE IF ANY PROGRAM CONSTRAINTS HAVE BEEN VIOLATED
 ISN 0003              DOUBLE PRECISION NAME
 ISN 0004              INTEGER PROG
 ISN 0005              LOGICAL SKIP,EXT,ACCL
 ISN 0006              INTEGER*2  NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,KODE,NYRSFX,
                      1  YDPL,KODEM,KODESP,
                      6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
               C
 ISN 0007              COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                      1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                      2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                      3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
                      4,  R(84),  S(84),CS(90),NPROG(90),KPROG(90),  KODE(90)
 ISN 0008              COMMON/SCRACH/M,N,NCS,PROG,IODD,IERR,SKIP,MYFLAG,JS,NSCALE(5),
                      1  NSL(10),TOTAL(20),W(20),D(20),XOUT(20),VOUT(20),RRR(20),YEAR(20)
                      2,  Y(20),KVEHI(50),LABEL(50),LVARY(70),LVD(70),IVEH(70),LVS(70),
                      3  LVSF(80),VNAM(80),NOP(86),RF(86),CF(86),SF(86),FLAGR(86),
                      4  FLAGS(86),NSSF(86),NSRF(86),NSXF(86),NDSF(86),SUSTF(86),NLVP(86)
                      5,  NSTRRC(86),NYRSRC(86),LNDF(86),NSTRST(86),LNDATE(86),NPRO(90),
                      6  KPRO(90),CSX(90),LZ(46),RCOST( 60), KVEH( 60),IMAGE(830),
                      7  XSCH(10,70),PLSCH(10,70),XLVSUM(20,50),RECUR(20,50),DUM(401)
               C
 ISN 0009              IERR = 0
 ISN 0010              IF (NCSTR.EQ.0) RETURN
 ISN 0012              NR = PROG
 ISN 0013              DO 100 I=1,NCSTR
 ISN 0014              J = NPROG(I)
 ISN 0015              K = KPROG(I)
 ISN 0016              IF (J.NE.NR.AND.K.NE.NR)  GO TO 100
 ISN 0018              MP= KODE(I)
 ISN 0019              IF (MP.LT.1.OR.MP.GT.11)  GO TO 100
 ISN 0021              GO TO (10,20,30,40,50,60,70,110,90,91,92),  MP
 ISN 0022           10 DT = CS(I)
 ISN 0023              IF (S(J).LT.(S(K)&R(K) & DT)) GO TO 110
 ISN 0025              GO TO 100
 ISN 0026           20 DT = CS(I)
 ISN 0027              IF ((S(J)&R(J)&DT).GT.S(K))  GO TO 110
 ISN 0029              GO TO 100
 ISN 0030           30     IF(S(J).NE.CS(I)) GO TO 110
 ISN 0032              GO TO 100
 ISN 0033           40 IF((S(J) & R(J) - 1.0).NE.CS(I))  GO TO 110
```

```
ISN 0035            GO TO 100
ISN 0036         50       IF (R(J).NE.CS(I)) GO TO 110
ISN 0038            GO TO 100
ISN 0039         60 DT = LNDATE(J)
ISN 0040            ET = LNDATE(K)
ISN 0041            IF((S(J)&DT&CS(I)).GT.(S(K)&ET).AND.(S(J) & DT & CS(I)).LT.(TREF &
                   1   20.))
                   1  GO TO 110
ISN 0043            GO TO 100
ISN 0044         70 DT = LNDATE(J) - 1
ISN 0045            IF ((S(J) & DT).GT.CS(I))   GO TO 110
ISN 0047            GO TO 100
ISN 0048         90  IF(S(J).LT.CS(I))  GO TO 110
ISN 0050            GO TO 100
ISN 0051         91 DT = LNDATE(J) - 1
ISN 0052            IF ((S(J) & DT).LT.CS(I))  GO TO 110
ISN 0054            GO TO 100
ISN 0055         92 DT = LNDATE(K) -1
ISN 0056            IF ((S(J)&R(J)&CS(I)).GT.(S(K)&DT))  GO TO 110
ISN 0058        100 CONTINUE
ISN 0059            RETURN
ISN 0060        110 IERR = 1
ISN 0061        120 RETURN
ISN 0062            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02CR(R)
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME
CONSTR	00	4F2							
SAV2	4F8	FE0							
SCRACH	14D8	6A60							

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
100	SAV2	SAV2	104	SCRACH	SCRACH
108	SCRACH	SCRACH	10C	SCRACH	SCRACH
110	SCRACH	SCRACH	114	SCRACH	SCRACH

```
ENTRY ADDRESS       00
TOTAL LENGTH      7F38

****MOX02CR   NOW REPLACED IN DATA SET
```

```
0001            SUBROUTINE DATINS
              C  ALL COST DATA ASSOCIATED WITH VEHICLES IS INPUT - ALSO MISSION DATA
0002            DOUBLE PRECISION NAME
0003            REAL NPERPD,LEVEL
0004            LOGICAL EXT,ACCL
0005            INTEGER*2   YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,KODE,
               1 NYRSFX,KODEM,KODESP,NU,NBY,MODE,NOB,FINISH,NSTG,NFML,NFMU,KODS,
               2  MAS,LABS,LABF,LABI,LSA,NYS,KODEF,LST,MST,IST,JST,KST,VEH,NYD,
               3  NMULT,NONREC,IS,MAT,LYR,LETT,LYD,MIN,NVS,MRV,NRP,NYP,KODEP,
               4  IVEHA,NTRIP,NPLS,NRR,MR,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,MAPS,
               5 MAPF,MAPI,KOUT,LTR,KODEV,NINTYR,NTGYTR,MAF,MAIC
              C
0006            COMMON/SAVER/  RFIXD(12,84)
0007            COMMON/SAVDMP/ NFAM,KFLAG,FAM(30),KODEF(30),FMNR(30),FMSUS(30),
               1JST(30),YDF(30),LSA(40),SNR(40),NYS(40),DINT(40),SINT(40),KST(40),
               2  YDI(40),YDS(40),IST(40),FMSLS(30,2),SUSLS(40,2),SINTLS(40,2),
               3  LST(30,5),YDPF(30,5),MST(30,10),YDPS(30,10)
0008            COMMON/SAVSAR/COR,POJ(3),SRJ(3,3),NU(40),NBY(40),NOB(40),RINT(40),
               1   PLCINT(40),XLT(40), PLCT(40),UPP(40),TAT(40),TAMT(50),SR(40,3),
               2   MODE(40,3),PLC(40,3)
0009            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
               1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
               2  RPLM(50),MAS(40,3), RXD(12,50)
0010            COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
               1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
               2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
               3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
               4,  P(84), S(84),CS(90),NPROG(90),KPROG(90), KODE(90)
0011            COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
               1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0012            COMMON/SAV4/  MAF(30,3), MAIC(40,3),
               *              NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
               1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
               2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
               3   PSTGS(30,10,2)
0013            COMMON/SVACAV/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
               1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
               2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
0014            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
               1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
               2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0015            COMMON/VARNCE/KSTAT,VARI(40),VARF(50),VARM(56),FMVAR(2,30),
               1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)

0016            COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
               1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
0017            COMMON/SCRACH/  IP,IV,IG,MODX(3),NFX(4),NPFAX(5),LSX(5),NPINXL(5),
               1  NPINXU(5),NPSTX(10),MSX(10),LZ(20),PB(50),MISN(50,20),DUM(1382),
               2  RCOST(60),RXM(50),II,KNSP,KLCK,IM,DUMM(3962),SRXX(3),XX(3),
               3  PX(56),CX(56),SX(56),TRX(56)
              C
0018            READ(5,100) LP,NOPT,MOS,NSOL,MSOL,MITR,ILY,MYRS,TREF,GUESS,GRO,
               1  SLO,COR,IP,IG,IFM,II,IM,ISD,IV
              C     ***IG IFM II IM ISD AND IV ARE VARIABLES FOR BATCHING ONLY ***
0019            IF (MYRS.EQ.0) GO TO 806
0020            WRITE (6,104)
0021            GRO = GRO/100.0
0022            IF(IG.LT.0)  GO TO 12
0023            WRITE(6,213)
0024            LX = 0
              C  NSDC = NUMBER OF SPECIAL DEVELOPMENT COSTS
              C       WARNING - DON'T USE NSDC WHILE BATCHING
0025            NSDC = 0
0026            DO 8000 I = 1,40
0027            READ(5,101)  KODX,   STG(I),(SR(I,J),J=1,3),(PLC(I,J),J=1,3),
               1  SNR(I),STS(I),LXX,NBX,(NFX(J),J=1,4),(MODX(J), J=1,3)
0028            IF(KODX.EQ.0)  GO TO 12
0029            KODS(I) = KODX
0030            LSA(I) = LXX
0031            NBY(I) = NBX
0032            DO 8010  J = 1,3
0033            NFS(I,J) = NFX(J)
0034         8010 MODE(I,J) = MODX(J)
0035            NFS(I,4) = NFX(4)
0036            IF(LCK.NE.1.AND.(PLC(I,1).GT..001.OR.PLC(I,2).GT..001.OR.PLC(I,3).
               1GT..001))   LCK = 1
              C  INPUT NU(I) LE -2 IF WANT PROGRAM TO CALCULATE ESTIMATE FOR NU
0037            READ (5,111)  (SUSLS(I,J),J=1,2),NUX,UPP(I),UPPXX,PXX,RPLO(I),
               1  YDS(I),ISX,NSXF,(SRXX(J),J=1,3),XX(1),SNRXX,XX(2),STSXX,XX(3)
0038            NU(I) = NUX
0039            IST(I) = ISX
0040            NSTG = NSTG + 1
0041            LABS(I) = 0
0042            NDS = YDS(I)
0043            NYS(I) = MAXO(IST(I) - ILY + NDS, 1)
0044            WRITE(6,112)
0045            WRITE (6,8001) STG(I),(SR(I,J),PLC(I,J),J=1,3),SNR(I),STS(I),
```

```
                1  NYS(I),LSA(I),(NFS(I,J),J=1,4),NBY(I)
0046             IF(XX(1).GT..0001)  CALL MEAN(XX(1),KSTAT,SVAR(1,I),SRXX(1),
                1  SR(I,1))
0047             IF(XX(1).GT..0001.AND.SR(I,2).GT..0001)
                1  CALL MEAN(XX(1),KSTAT,SVAR(2,I),SRXX(2),SR(I,2))
0048             IF(XX(1).GT..0001.AND.SR(I,3).GT..0001)
                1  CALL MEAN(XX(1),KSTAT,SVAR(3,I),SRXX(3),SR(I,3))
0049             IF(XX(2).GT..0001)  CALL MEAN(XX(2),KSTAT,SVAR(4,I),SNRXX,SNR(I))
0050             IF(XX(3).GT..0001) CALL MEAN(XX(3),KSTAT,SVAR(5,I),STSXX,STS(I))
0051             UP = UPP(I)
0052           . IF(PXX.GT..0001) CALL MEAN(PXX,KSTAT,VARI(I),UPPXX,UPP(I))
0053             IF(XX(1).GT..001.OR.XX(2).GT..001.OR.XX(3).GT..001)
                *WRITE (6,8001) STG(I),(SR(I,J),PLC(I,J),J=1,3),SNR(I),STS(I),
                1  NYS(I),LSA(I),(NFS(I,J),J=1,4),NBY(I)
0054             DO 8002 J = 1,3
0055             IF (MODE(I,J).EQ.0) GO TO 8002
0056             LX = LX + 1
0057             MODE(I,J) = LX
0058             READ(5,8003) (SRJ(LX,K), K = 1,3), POJ(LX),SRJXX,PXX
0059             IF(PXX.LT..0001)  GO TO 8011
0060             CALL MEAN(PXX,KSTAT,SVAR(J,I),SRJXX,SRJ(LX,1))
0061             SRJ(LX,2) = .5*SRJ(LX,2)*(EXP(1.5*SVAR(J,I)) + 1.0)
0062             SRJ(LX,3) = .5*SRJ(LX,3)*(EXP(1.5*SVAR(J,I)) + 1.0)
0063        8011 WRITE(6,8004) J,POJ(LX),SRJ(LX,1),POJ(LX),(SRJ(LX,K),K=2,3)
0064        8002 CONTINUE
0065             IF(NU(I).NE.0)  WRITE(6,8005)  UPP(I),UP
0066             IF(NSXF.EQ.0)   GO TO 8000
0067             NSDC = NSDC + 1
0068             READ(5,110) NRXF, (RXD(J,NSDC), J = 1,12),RXDXX,PXX
0069             NRFX(NSDC) = NRXF
0070             NSFX(NSDC) = NSXF
0071             NX = NRFX(NSDC) + NSFX(NSDC) - ILY
0072             NDS = NYS(I)
0073             NYS(I) = MAXO(NDS,NX)
0074             LABS(I) = NSDC
0075             WRITE(6,113) (RXD(J,NSDC), J = 1,12)
0076             IF(PXX.LT..0001)  GO TO 8000
0077             RX = 0.0
0078             DO 500 J = 1,12
0079         500 RX = RX + RXD(J,NSDC)
0080             CALL MEAN(PXX,KSTAT,VARF(NSDC),RXDXX,RX)
0081             DO 501 J = 1,12
0082         501 RXD(J,NSDC) = .5*RXD(J,NSDC)*(EXP(1.5*VARF(NSDC))+ 1.0)


0083             WRITE(6,113) (RXD(J,NSDC), J = 1,12)
0084        8000 CONTINUE
0085          12 IF(IFM.LT.0)  GO TO 14
0086             DO 13 J = 1,30
0087             READ(5,102) I,FAM(I),FMNR(I),FMSUS(I),YDF(I),JX,NSXF,
                1  (FMSLS(I,K),K=1,2),FMNRXX,XX(1),FMSSXX,XX(2)
0088             IF(I.EQ.0)  GO TO 14
0089             IF(J.EQ.1)  WRITE(6,214)
0090             NFAM = NFAM + 1
0091             JST(I) = JX
0092             LABF(I) = 0
0093             KODEF(J) = I
0094             WRITE(6,112)
0095             WRITE(6,2141)KODEF(J), FAM(I),FMNR(I),FMSUS(I)
0096             IF(XX(1).GT..0001)  CALL MEAN(XX(1),KSTAT,FMVAR(1,I),FMNRXX,
                1    FMNR(I))
0097             IF(XX(2).GT..0001) CALL MEAN(XX(2),KSTAT,FMVAR(2,I),FMSSXX,FMSUS(I
                1))
0098             IF(XX(1).GT..001.OR.XX(2).GT..001)
                1WRITE(6,2141)KODEF(J), FAM(I),FMNR(I),FMSUS(I)
0099             IF(NSXF.EQ.0)  GO TO 13
0100             NSDC = NSDC + 1
0101             READ(5,110)   NRXF,     (RXD(J1,NSDC),  J1=1,12),RXDXX,PXX
0102             NSFX(NSDC) = NSXF
0103             NRFX(NSDC) = NRXF
0104             LABF(I) = NSDC
0105             WRITE(6,113) (RXD(J1,NSDC), J1 = 1,12)
0106             IF(PXX.LT..0001)  GO TO 13
0107             RX = 0.0
0108             DO 504 J1 = 1,12
0109         504 RX = RX + RXD(J1,NSDC)
0110             CALL MEAN(PXX,KSTAT,VARF(NSDC),RXDXX,RX)
0111             DO 505 J1 = 1,12
0112         505 RXD(J1,NSDC) = .5*RXD(J1,NSDC)*(EXP(1.5*VARF(NSDC)) + 1.0)
0113             WRITE(6,113) (RXD(J1,NSDC), J1 = 1,12)
0114          13 CONTINUE
0115          14 IF(II.LT.0)  GO TO 1716
0116             DO 1715 I = 1,40
0117             READ(5,103)  J,K,RINT(I),PLCINT(I),DINT(I),SINT(I),YDI(I),KX,NSXF,
                1    (SINTLS(I,L),  L=1,2)
0118             IF(J.EQ.0)  GO TO 1716
0119             IF(I.EQ.1)   WRITE(6,215)
0120             READ(5,108) RINTXX,XX(1),DINTXX,XX(2),SINTXX,XX(3)
```

```
0121              IF(LCK.NE.1.AND.PLCINT(I).GT..001)  LCK = 1
0122              NCI = NCI + 1
0123              LABI(I) = 0
0124              KST(I) = KX
0125              NFML(I) = J
0126              NFMU(I) = K
0127              WRITE(6,112)
0128              WRITE(6,216) FAM(J),FAM(K),RINT(I),PLCINT(I),DINT(I),SINT(I)
0129              IF(XX(1).GT..0001)  CALL MEAN(XX(1),KSTAT,FIVAR(1,I),RINTXX,
                 1   RINT(I))
0130              IF(XX(2).GT..0001)  CALL MEAN(XX(2),KSTAT,FIVAR(2,I),DINTXX,
                 1  DINT(I))
0131              IF(XX(3).GT..00001)  CALL MEAN(XX(3),KSTAT,FIVAR(3,I),SINTXX,
                 1  SINT(I))
0132              IF(XX(1).GT..001.OR.XX(2).GT..001.OR.XX(3).GT..001)
                 1WRITE(6,216) FAM(J),FAM(K),RINT(I),PLCINT(I),DINT(I),SINT(I)
0133              IF(NSXF.EQ.0)   GO TO 1715
0134              NSDC = NSDC + 1
0135              READ(5,110)  NRXF,       (RXD(J,NSDC), J = 1,12),RXDXX,PXX
0136              NRFX(NSDC) = NRXF
0137              NSFX(NSDC) = NSXF
0138              LABI(I) = NSDC
0139              WRITE(6,113) (RXD(J,NSDC), J = 1,12)
0140              IF(PXX.LT..0001)  GO TO 1715
0141              RX = 0.0
0142              DO 502 J = 1,12
0143          502 RX = RX + RXD(J,NSDC)
0144              CALL MEAN(PXX,KSTAT,VARF(NSDC),RXDXX,RX)
0145              DO 503 J = 1,12
0146          503 RXD(J,NSDC) = .5*RXD(J,NSDC)*(EXP(1.5*VARF(NSDC)) + 1.0)
0147              WRITE(6,113) (RXD(J,NSDC), J = 1,12)
0148         1715 CONTINUE
0149         1716 IF (IP.LT.0) GO TO 9002
0150              DO 9004 I = 1,30
0151              READ(5,9005)  J1, PAD(I),NPERPD(I)
0152              IF(J1.EQ.0)  GO TO 9002
0153              IF(I.EQ.1)  WRITE (6,9003)
0154              KODEP(I) = J1
0155              WRITE (6,9006) KODEP(I),PAD(I),NPERPD(I)
0156              NP = NP + 1
0157              READ(5,5000)  (NPSTX(J),PSTGD(I,J,1),YDPS(I,J),MSX(J),PSTGS(I,J,1)
                 1,  PSTGD(I,J,2),PSTGS(I,J,2),   J=1,10)
0158              DO 700 J = 1,10
0159              MST(I,J) = MSX(J)
0160          700 NPSTG(I,J) = NPSTX(J)
0161              READ(5,5000)  (NPFAX(J),PFAMD(I,J,1),YDPF(I,J),LSX(J),PFAMS(I,J,1)
                 1,   PFAMD(I,J,2),PFAMS(I,J,2),  J=1,5)
0162              READ(5,5002) (NPINXL(J),NPINXU(J),(PINTS(I,J,K),K=1,2),J=1,5)
0163              DO 701 J = 1,5
0164              LST(I,J) = LSX(J)
0165              NPFAM(I,J) = NPFAX(J)
0166              NPINTL(I,J) = NPINXL(J)
0167          701 NPINTU(I,J) = NPINXU(J)
0168              DO 9022 J = 1,10
0169              IF (NPSTG(I,J).EQ.0) GO TO 5009
0170              DO 9023 L = 1,NSTG
0171              IF (NPSTG(I,J).NE.KODE(L)) GO TO 9023
0172              NPSTG(I,J) = L
0173              WRITE(6,5003) STG(L),(PSTGD(I,J,K),PSTGS(I,J,K),K=1,3)
0174              GO TO 9022
0175         9023 CONTINUE
0176         9022 CONTINUE
0177         5009 DO 5006 J = 1,5
0178              IF (NPFAM(I,J).EQ.0) GO TO 5007
0179              L = NPFAM(I,J)
0180         5006 WRITE(6,5004) FAM(L),(PFAMD(I,J,K),PFAMS(I,J,K),K=1,3)
0181         5007 DO 5008 J = 1,5
0182              IF (NPINTL(I,J).EQ.0) GO TO 9004
0183              L = NPINTL(I,J)
0184              LX = NPINTU(I,J)
0185         5008 WRITE(6,5005) FAM(L),FAM(LX),(PINTS(I,J,K),K = 1,3)
0186         9004 CONTINUE
0187         9002 IF(IM.LT.0)  GO TO 19
0188              DO 1719 I=1,MYRS
0189         1719 LZ(I)=ILY+I-1
0190              WRITE(6,217) (LZ(I),I=1,MYRS)
0191         1717 DO 1718 I=1,50
0192              READ(5,105)  KM,NAME(I),PB(I),NSYX,NYRSXF,VLR(I),RPLM(I),TAMT(I),
                1  WPR(I),NTP,(MISN(I,J), J=1,MYRS)
0193              IF(KM.EQ.0)   GO TO 1720
0194              KODEM(I) = KM
0195              NSYR(I) =  NSYX
0196              NYRSFX(I) = NYRSXF
0197              NTRIP(I) = NTP
0198              NMIS = NMIS + 1
0199              READ(5,107) PLR(I),SUS(I),C(I),NDPL,(RDIST(I,L),L=1,4),NPS,MRX,
```

```
     1    LRX,NR,IIS,IVAX
0200      YDPL(I) = NDPL
0201      NPLS(I) = NPS
0202      MR(I) = MRX
0203      LTR(I) = MAXO(LRX,1)
0204      NRR(I) = NR
0205      IS(I) = 1900 + IIS
0206      IVEHA(I) = IVAX
0207      WRITE(6,219) I,NAME(I),  VLR(I),WPR(I),PB(I),LTR(I),
     1  (MISN(I,J),J=1,MYRS)
0208      PX(I) = PLR(I)
0209      CX(I) = C(I)
0210      SX(I) = SUS(I)
0211      READ(5,108) PLRXX,XX(1),CXX,XX(2),SUSXX,XX(3)
0212      IF(XX(1).GT..0001) CALL MEAN(XX(1),KSTAT,PLVAR(1,I),PLRXX,PLR(I))
0213      IF(XX(2).GT..0001) CALL MEAN(XX(2),KSTAT,PLVAR(2,I),CXX,C(I))
0214      IF(XX(3).GT..0001) CALL MEAN(XX(3),KSTAT,PLVAR(3,I),SUSXX,SUS(I))
0215      RXM(I)  = 0.0
0216      TRX(I) = 0.0
0217      IF(NYRSFX(I).EQ.0)  GO TO 1718
0218      READ(5,110)  NSTRXF,  (RFIXD(J,I),  J=1,12),RXDXX,PXX
0219      NSTRFX(I) = NSTRXF
0220      DO 520 J = 1,12
0221  520 RXM(I) = RXM(I) + RFIXD(J,I)
0222      TRX(I) = RXM(I)
0223      IF(PXX.LT..0001)  GO TO 1718
0224      CALL MEAN(PXX,KSTAT,VARM(I),RXDXX,RXM(I))
0225      DO 522 J = 1,12
0226  522 RFIXD(J,I) = .5*RFIXD(J,I)*(EXP(1.5*VARM(I)) + 1.0)
0227 1718 CONTINUE
0228 1720 WRITE(6,104)
0229      DO 1721 I = 1,NMIS
0230      WRITE(6,112)
0231      IF(KSTAT.GT.0)  WRITE(6,109) I,NAME(I),PX(I), (RDIST(I,L),L=1,4),
     *  CX(I),YDPL(I),IS(I),SX(I),TRX(I)
0232 1721 WRITE(6,109) I,NAME(I),PLR(I),(RDIST(I,L),L=1,4),C(I),YDPL(I),
     1  IS(I),SUS(I),RXM(I)
0233   19 IF(ISD.LT.0)  GO TO 20
     C   INPUT SPECIAL PROGRAMS HAVING NO ASSOCIATED LAUNCHES
     C   KODESP GT 100
0234      DO 1725 I = 1,6
0235      K = NMIS + I
0236      READ(5,106)  KO,NAME(K),C(K),NDPL,IIS,SUS(K),NST,NYRSXF,CXX,XX(I),
     1  SUSXX,XX(2)
0237      IF(KO.EQ.0)  GO TO 20
0238      KODESP(I) = KO
0239      YDPL(K) = NDPL
0240      IS(K)      = 1900 + IIS
0241      NYRSST(K)      = NST
0242      NYRSFX(K)      = NYRSXF
0243      NSPR = NSPR + 1
0244      IF(I.EQ.1)  WRITE(6,114)
0245      WRITE(6,112)
0246      WRITE(6,115) I,NAME(K),C(K),SUS(K),IS(K),NDPL
0247      IF(XX(1).GT..0001) CALL MEAN(XX(1),KSTAT,PLVAR(2,K),CXX,C(K))
0248      IF(XX(2).GT..0001)  CALL MEAN(XX(2),KSTAT,PLVAR(3,K),SUSXX,SUS(K))
0249      IF(XX(1).GT..001.OR.XX(2).GT..001)
     *WRITE(6,115) I,NAME(K),C(K),SUS(K),IS(K),NDPL
0250      IF(NYRSXF.EQ.0)  GO TO 1725
0251      READ(5,110)  NSTRXF,  (RFIXD(J,K),      J=1,12),RXDXX,PXX
0252      NSTRFX(K)      = NSTRXF
0253      WRITE(6,113)  (RFIXD(J,K), J = 1,12)
0254      IF(PXX.LT..0001)   GO TO 1725
0255      RX = 0.0
0256      DO 521 J = 1,12
0257  521 RX = RX + RFIXD(J,K)
0258      CALL MEAN(PXX,KSTAT,VARM(K),RXDXX,RX)
0259      DO 523 J = 1,12
0260  523 RFIXD(J,K) = .5*RFIXD(J,K)*(EXP(1.5*VARM(K)) + 1.0)
0261      WRITE(6,113)  (RFIXD(J,K), J = 1,12)
0262 1725 CONTINUE
0263   20 IF(IG.LT.0)  NSTG = KNSTG
0264      IF(IG.LT.0)  LCK = KLCK
0265      IF(IFM.LT.0)  NFAM = KNFAM
0266      IF(IQ.LT.0)  NCI = KNCI
0267      IF (IP.LT.0) NP=KNP
0268      IF(ISD.LT.0)  NSPR=KNSP
0269      IF(IM.LT.0)  NMIS = KNMIS
0270      RETURN
0271  806 WRITE(6,4102)
0272   99 RETURN
0273  100 FORMAT (8I3,F5.1,F12.2,F3.1,F5.1,F4.2,13X,712)
0274  101 FORMAT (I2,1X,A4,3F6.3,3F5.3,3X,  2F6.3,3X,6I3,1X,3I1)
0275  102 FORMAT(I2,1X,A4,2F10.0,F4.1,2I3,2F10.0,F7.0,F3.2,F7.0,F3.2)
0276  103 FORMAT (2X,2I3,4F10.0,F4.1,2I3,2F10.0)
0277  104 FORMAT (1H1)
```

```
0278          105 FORMAT (I2,A6,F4.2,2X,2I2,F7.0,2F3.0,F7.0,I2,20I2)
0279          106 FORMAT(I3,   A6,F10.2,I5,I2,F10.2,2I2,F10.2,F3.2,F10.2,F3.2)
0280          107 FORMAT (2X,3F10.2,I5,  4F5.3,  10X,5I2,I3)
0281          108 FORMAT (3(F10.0,F3.2))
0282          109 FORMAT(1X,I2,1X,A6,2X,4HPLR=,F6.1,8H DIST BY,4(F3.2,2H, ),5H DEV=,
                 1  F8.1,
                 2 5H  FOR,I5,14H YRS  STARTING,I6,2X,5HSUST=,F8.1,3X,6HFIXED=,F8.1)
0283          110 FORMAT (I3,12F5.2,F6.1,F3.2)
0284          111 FORMAT (4X,2F5.0,I3,F6.2,F6.1,F3.2,F6.0,F2.0,2I2,3F5.1,F3.2,F6.1,
                *F3.2,F6.1,F3.2)
0285          112 FORMAT (1H )
0286          113 FORMAT (14H FIXED COSTS =, 12F9.2)
0287          114 FORMAT (1H1,4X,'SPECIAL PROGRAMS'//)
0288          115 FORMAT(1H ,I3,1X,A6,3X,'DEV =',F8.2,3X,'SUST =',F8.2,3X,
                 *  'DEV STARTS', I6,3X,'FOR ',I4,1X,'YEARS')
0289          213 FORMAT (16H STAGE COST DATA/6HOTITLE,3(16H  RECURRING  LC ),68H  D
                 1EVELOPMENT  SUSTAINING  AVAILABLE  SHARED COST GROUPS  BATCH FACT/
                 2  10X,10H(HARDWARE),
                 3        6X,10H(ETR ONLY),6X,10H(WTR ONLY),30X,8HFROM  TO//)
0290          214 FORMAT (1HO///17HOSHARED COST DATA/37HONO. TITLE  DEVELOPMENT  S
                 1USTAINING//)
0291          215 FORMAT (1HO///22HOINTEGRATION COST DATA/59HOLOWER  UPPER  RECUR
                 1RING  LC  DEVELOPMENT  SUSTAINING/14H GROUP  GROUP//)
0292          216 FORMAT (2X,A4,4X,A4,F11.2,F7.3,2F13.2)
0293          217 FORMAT (14H1MISSION MODEL/48HO  MISSION  VELOCITY  PAYLOAD  P
                 1RIORITY  TR, 17X, 15HLAUNCH SCHEDULE//50X,2014/1H /)
0294          219 FORMAT (1X,I2,1X,A6,2X,2F10.0,F10.2,4X,I2,2X,2014)
0295         2141 FORMAT (1X,I2,2X,A4,2X,2F13.2)
0296         4102 FORMAT (1HO///5X,26HEND OF DATA - JOB COMPLETE)
0297         5000 FORMAT(2(2X,I2,F5.0,F3.0,I3,3F5.0,10X))
0298         5002 FORMAT(8X,2I3,2F6.0,6X,2I3,2F6.0,6X,2I3,2F6.0,6X)
0299         5003 FORMAT (27X,A4,1X,5HSTAGE,17X,3(F9.2,F8.2))
0300         5004 FORMAT (27X,A4,1X,6HSHARED,16X,3(F9.2,F8.2))
0301         5005 FORMAT (27X,15HINTEGRATION OF ,A4,5H AND ,A4,8X,F8.2,2(9X,F8.2))
0302         8001 FORMAT
                 1  (1X,A4,1X,3(F9.2,F7.3),F13.2,F12.2,2X,I4,1X,I4,2X,4I4,I9)
0303         8003 FORMAT (4X,5F10.3,F3.2)
0304         8004 FORMAT (3X,19HRECURRING COST TYPE,I2,22H  FOR X LESS THAN OR =,
                 1 F6.2,14H, TOTAL COST =,F6.2,19H.FOR X GREATER THAN,F6.2,
                 2 14H, TOTAL COST =,F6.2,4H X +,F6.2)
0305         8005 FORMAT(16HO REUSABLE STAGE, 4X,20HUNIT PURCHASE PRICE=,F7.2,'  ('
                 *  F7.2, ')')
0306         9003 FORMAT(1HO///14HOPAD COST DATA/12HONO. COMPLEX,2X,11HLAUNCHES/YR,
                 1  37X,5HPAD 1,12X,
                 2  5HPAD 2,12X,5HPAD 3/59X,3(11HDEV    SUST,6X)//)
0307         9005 FORMAT (I4,2X,A4,F5.0)
0308         9006 FORMAT (1X,I2,2X,A4,5X,F6.2)
0309              END
```

TOTAL MEMORY REQUIREMENTS 003ED0 BYTES

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP          DEFAULT OPTION(S) USED
         VARIABLE OPTIONS USED - SIZE=(126976,24576)
IEW0000      NAME MOX02DS(R)
IEW0461  IBCOM=
IEW0461  MEAN
IEW0461  MAXO
IEW0461  EXP

                                                    MODULE MAP

  CONTROL SECTION                                ENTRY
    NAME     ORIGIN   LENGTH                     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
  DATINS       00      3ED0
  SAVER       3ED0      FC0
  SAVDMP      4E90     148C
  SAVSAR      6350      A5C
  SAVE1       6DB0      FC4
  SAV2        7D78      FE0
  SAV3        8D58      980
  SAV4        96D8     3188
  SVACAV      C860      B48
  SAVALL      D3A8     3A1C
  VARNCE      10DC8     A0C
  TEMP        11BA8     4110
  SCRACH      15988     6A60

  ENTRY ADDRESS       00
  TOTAL LENGTH     1C418

****MOX02DS   NOW REPLACED IN DATA SET

(17)          OS/360  FORTRAN H                                  DATE  71.312/17.10.48

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
   ISN 0002             SUBROUTINE DECSNI
                C
                C  THIS SUBROUTINE SETS UP DS COSTS, CALCULATES AVAILABILITY OF EACH DECISION
                C   COST, AND MATCHES THESE COSTS WITH EACH VEHICLE THEN PRINTS THEM OUT
                C
   ISN 0003             REAL NPERPD
   ISN 0004             INTEGER*2 LTR,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,MAPS,MAPF,MAPI,
                       1  FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,LSA,NYS,KODEF,
                       2  LST,MST,IST,JST,KST,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,
                       3  MIN,MAF,MAIC
   ISN 0005             COMMON/SAVDMP/ NFAM,KFLAG,FAM(30),KODEF(30),FMNR(30),FMSUS(30),
                      1JST(30),YDF(30),LSA(40),SNR(40),NYS(40),DINT(40),SINT(40),KST(40),
                       2  YDI(40),YDS(40),IST(40),FMSLS(30,2),SUSLS(40,2),SINTLS(40,2),
                       3  LST(30,5),YDPF(30,5),MST(30,10),YDPS(30,10)
   ISN 0006             COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                      1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                       2  RPLM(50),MAS(40,3), RXD(12,50)
   ISN 0007             COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                       1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
   ISN 0008             COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                       *             NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                       1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                       2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                       3   PSTGS(30,10,2)
   ISN 0009             COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                      1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                       2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
   ISN 0010             COMMON/SCRACH/LYF(30),NYF(30),DUM(6748)
                C
   ISN 0011             IF(FINISH.GT.1)  GO TO 2
                C
                C      ***SET UP DS  COSTS FOR BRANCH AND BOUND PROCEDURE***
                C      CALCULATE AVAILABILITY OF EACH DECISION COST
                C
   ISN 0013             NUMD = 0
   ISN 0014             DO 3 I = 1,NSTG
   ISN 0015             IF(LSA(I).GT.MYRS)  LSA(I) = MYRS
   ISN 0017             MAS(I,1) = 0
   ISN 0018             X = LABS(I)
   ISN 0019             IF(SNR(I)&STS(I)&X.LT..01)  GO TO 9024
   ISN 0021             NUMD = NUMD & 1
   ISN 0022             DS (NUMD)=SNR(I)
```

```
ISN 0023            IF(LABS(I).EQ.0)  GO TO 302
ISN 0025            L = LABS(I)
ISN 0026            DO 301 K = 1,12
ISN 0027          301 DS(NUMD) = DS(NUMD) & RXD(K,L)
ISN 0028          302 SUST (NUMD)=STS(I)
ISN 0029            MAT(NUMD) = I
ISN 0030            MAS(I,1) = NUMD
                  C. NYD = FIRST YEAR COMPONENTS DEPENDENT ON DEV COST NUMD ARE AVAILABLE
ISN 0031            NYD(NUMD) = NYS(I)
                  C  LYD = LAST YEAR COMPONENTS DEPENDENT ON DEV COST NUMD ARE AVAILABLE
ISN 0032            LYD(NUMD) = LSA(I)
ISN 0033            YD(NUMD) = YDS(I)
ISN 0034            NDUM = NUMD & NMIS & NSPR
ISN 0035            IS(NDUM) = IST(I) & 1900
ISN 0036          9024 DO 9025 J = 1,2
ISN 0037            MAS(I,J&1) = 0
ISN 0038            IF(SUSLS(I,J).LT..01.OR.NP.EQ.0)    GO TO 9025
ISN 0040            NUMD = NUMD & 1
ISN 0041            DS(NUMD) = 0.0
ISN 0042            SUST(NUMD) = SUSLS(I,J)
ISN 0043            MAT(NUMD) = I & 2000
ISN 0044            MAS(I,J&1) = NUMD
ISN 0045            NYD(NUMD) = NYS(I)
ISN 0046            LYD(NUMD) = LSA(I)
ISN 0047            YD(NUMD) = 0.0
ISN 0048            NDUM = NUMD & NMIS & NSPR
ISN 0049            IS(NDUM) = ILY & 1900
ISN 0050          9025 CONTINUE
ISN 0051             3 CONTINUE
ISN 0052            IF(NFAM.EQ. 0) GO TO 601
                  C      CALCULATE FAMILY AVAILABILITY DATE
                  C FIRST YR. FAMILY IS AVAIL. = 1ST YR. ANY STAGE IN THAT FAMILY IS AVAIL.
ISN 0054            DO 422 I1 = 1,NFAM
ISN 0055            I = KODEF(I1)
ISN 0056            LYF(I) = 0
ISN 0057          422 NYF(I) = MYRS
ISN 0058            DO 423 J = 1,NSTG
ISN 0059            DO 424 MS = 1,4
ISN 0060            I = NFS(J,MS)
ISN 0061            IF(I.EQ.0)  GO TO 423
ISN 0063            IF(NYF(I).GT.NYS(J))  NYF(I) = NYS(J)
ISN 0065            IF(LYF(I).LT.LSA(J))  LYF(I) = LSA(J)

ISN 0067          424 CONTINUE
ISN 0068          423 CONTINUE
ISN 0069            DO 425 I1 = 1,NFAM
ISN 0070            I = KODEF(I1)
ISN 0071            IF(YDF(I).LE.0.01.OR.NYF(I).EQ.1)   GO TO 425
ISN 0073            NX = YDF(I) & .9
ISN 0074            NX = MAXO(JST(I) - ILY & NX,1)
ISN 0075          *   IF(NYF(I).GT.NX)  NYF(I) = NX
ISN 0077          425 CONTINUE
ISN 0078            DO 6 I1 = 1,NFAM
ISN 0079            I = KODEF(I1)
ISN 0080            MAF(I,1) = 0
ISN 0081            X = LABF(I)
ISN 0082            IF(FMNR(I)&FMSUS(I)&X.LT..01)  GO TO 9026
ISN 0084            NUMD = NUMD & 1
ISN 0085            DS (NUMD) = FMNR(I)
ISN 0086            IF(LABF(I).EQ.0)  GO TO 304
ISN 0088            L = LABF(I)
ISN 0089            DO 303 K = 1,12
ISN 0090          303 DS(NUMD) = DS(NUMD) & RXD(K,L)
ISN 0091          304 SUST (NUMD)=FMSUS(I)
ISN 0092            MAT(NUMD) = -I
ISN 0093            MAF(I,1) = NUMD
ISN 0094            NYD(NUMD) = NYF(I)
ISN 0095            LYD(NUMD) = LYF(I)
ISN 0096            YD(NUMD) = YDF(I)
ISN 0097            NDUM = NUMD & NMIS & NSPR
ISN 0098            IS(NDUM) = JST(I) & 1900
ISN 0099          9026 DO 9027 J = 1,2
ISN 0100            MAF(I,J&1) = 0
ISN 0101            IF (FMSLS(I,J).LT..01.OR.NP.EQ.0)   GO TO 9027
ISN 0103            NUMD = NUMD & 1
ISN 0104            DS(NUMD) = 0.0
ISN 0105            SUST(NUMD) = FMSLS(I,J)
ISN 0106            MAT(NUMD) = -1 & 2000
ISN 0107            MAF(I,J&1) = NUMD
ISN 0108            NYD(NUMD) = NYF(I)
ISN 0109            LYD(NUMD) = LYF(I)
ISN 0110            YD(NUMD) = 0.0
ISN 0111            NDUM = NUMD & NMIS & NSPR
ISN 0112            IS(NDUM) = ILY & 1900
ISN 0113          9027 CONTINUE
```

```
ISN 0114          6 CONTINUE
ISN 0115        601 IF(NCI.EQ.0) GO TO 61
ISN 0117            DO 60 I = 1,NCI
ISN 0118            JF = NFML(I)
ISN 0119            KF = NFMU(I)
ISN 0120            MAIC(I,1) = 0
ISN 0121            X = LARI(I)
ISN 0122            IF(DINT(I)&SINT(I)&X.LT..01)  GO TO 9028
ISN 0124          .  NUMD = NUMD & 1
ISN 0125            DS (NUMD) = DINT(I)
ISN 0126            IF(LABI(I).EQ.0)  GO TO 306
ISN 0128            L = LARI(I)
ISN 0129            DO 305 K = 1,12
ISN 0130        305 DS(NUMD) = DS(NUMD) & RXD(K,L)
ISN 0131        306 SUST (NUMD)=SINT(I)
ISN 0132            MAT(NUMD) = -100 -I
ISN 0133            MAIC(I,1) = NUMD
                C FIRST YR. INT. COST IS AVAIL. = 1ST YR. BOTH FAMS. ARE AVAIL.
ISN 0134            NYD(NUMD) = MAXO(NYF(JF),NYF(KF))
ISN 0135            LYD(NUMD) = MINO(LYF(JF),LYF(KF))
ISN 0136            IF(YDI(I).LE.0.01.OR.NYD(NUMD).EQ.1)  GO TO 307
ISN 0138            NX = YDI(I) & .9
ISN 0139            NX = MAXO(KST(I) & NX - ILY,1)
ISN 0140            IF(NYD(NUMD).GT.NX)  NYD(NUMD) = NX
ISN 0142        307 YD(NUMD) = YDI(I)
ISN 0143            NDUM = NUMD & NMIS & NSPR
ISN 0144            IS(NDUM) = KST(I) & 1900
ISN 0145        9028 DO 9029 J = 1,2
ISN 0146            MAIC(I,J&1) = 0
ISN 0147            IF (SINTLS(I,J).LT..01.OR.NP.EQ.0)    GO TO 9029
ISN 0149            NUMD = NUMD & 1
ISN 0150            DS(NUMD) = 0.0
ISN 0151            SUST(NUMD) = SINTLS(I,J)
ISN 0152            MAT(NUMD) = -100 - I & 2000
ISN 0153            MAIC(I,J&1) = NUMD
ISN 0154            NX = YDI(I) & .9
ISN 0155            NX = MAXO( KST(I) & NX - ILY, 1)
ISN 0156            NNX = MAXO(NYF(JF),NYF(KF))
ISN 0157            NYD(NUMD) = MINO(NNX,NX)
ISN 0158            LYD(NUMD) = MINO(LYF(JF),LYF(KF))
ISN 0159            YD(NUMD) = 0.0
ISN 0160            NDUM = NUMD & NMIS & NSPR

ISN 0161            IS(NDUM) = ILY & 1900
ISN 0162        9029 CONTINUE
ISN 0163          60 CONTINUE
ISN 0164          61 IF (NP.EQ.0) GO TO 9010
ISN 0166            DO 9011 I = 1,NP
ISN 0167            DO 9030 J = 1,5
ISN 0168            MAPF(I,J) = 0
ISN 0169            IF(PFAMD(I,J,1) & PFAMS(I,J,1) .LT. .01) GO TO 9030
ISN 0171            NUMD = NUMD & 1
ISN 0172            DS(NUMD) = PFAMD(I,J,1)
ISN 0173            SUST(NUMD) = PFAMS(I,J,1)
ISN 0174            MAT(NUMD) = -200 - I & 2000
ISN 0175            MAPF(I,J) = NUMD
ISN 0176            NX = YDPF(I,J) & .9
ISN 0177            NX = NX & LST(I,J) - ILY
ISN 0178            NYD(NUMD) = MAXO (NX,1)
ISN 0179            LYD(NUMD) = MYRS
ISN 0180            YD(NUMD) = YDPF(I,J)
ISN 0181            NDUM = NUMD & NMIS & NSPR
ISN 0182            IS(NDUM) = LST(I,J) & 1900
ISN 0183        9030 CONTINUE
ISN 0184            DO 9031 J = 1,10
ISN 0185            MAPS(I,J) = 0
ISN 0186            IF (PSTGD(I,J,1) & PSTGS(I,J,1).LT..01 ) GO TO 9031
ISN 0188            NUMD = NUMD & 1
ISN 0189            DS(NUMD) = PSTGD(I,J,1)
ISN 0190            SUST(NUMD) = PSTGS(I,J,1)
ISN 0191            MAT(NUMD) = -300 - I & 2000
ISN 0192            MAPS(I,J) = NUMD
ISN 0193            NX = YDPS(I,J) & .9
ISN 0194            NX = NX & MST(I,J) - ILY
ISN 0195            NYD(NUMD) = MAXO (NX,1)
ISN 0196            LYD(NUMD) = MYRS
ISN 0197            YD(NUMD) = YDPS(I,J)
ISN 0198            NDUM = NUMD & NMIS & NSPR
ISN 0199            IS(NDUM) = MST(I,J) & 1900
ISN 0200        9031 CONTINUE
ISN 0201            DO 9032 J = 1,5
ISN 0202            MAPI(I,J) = 0
ISN 0203            IF (PINTS(I,J,1) .LT. .01) GO TO 9032
ISN 0205            NUMD = NUMD & 1
ISN 0206            DS(NUMD) = 0.0
```

```
ISN 0207            SUST(NUMD) = PINTS(I,J,1)
ISN 0208            MAT(NUMD) = -400 - I & 2000
ISN 0209            MAPI(I,J) = NUMD
ISN 0210            JF = NPINTL(I,J)
ISN 0211            KF = NPINTU(I,J)
ISN 0212            NYD(NUMD) = MAXO(NYF(JF),NYF(KF))
ISN 0213            LYD(NUMD) = MYRS
ISN 0214            YD(NUMD) = 0.0
ISN 0215            NDUM = NUMD & NMIS & NSPR
ISN 0216            IS(NDUM) = ILY & 1900
ISN 0217         9032 CONTINUE
ISN 0218         9011 CONTINUE
                C
ISN 0219         9010 CALL MATCHI
                C
ISN 0220             IF(KFLAG.EQ.1)  RETURN
                C
ISN 0222             2 CALL PRINTI
                C
ISN 0223             RETURN
ISN 0224             END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOX02DN(R)
IEW0461  MATCHI
IEW0461  PRINTI
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH
DECSNI	00	11FE
SAVDMP	1200	148C
SAVE1	26C0	FC4
SAV3	3688	980
SAV4	4008	3188
SAVALL	7190	3A1C
SCRACH	ABB0	6A60

ENTRY NAME LOCATION NAME LOCATION NAME LOCATION NAME

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
128	SAVDMP	SAVDMP
130	SAV3	SAV3
138	SAV4	SAV4
140	SAVALL	SAVALL
148	SCRACH	SCRACH
150	PRINTI	$UNRESOLVED

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
12C	SAVE1	SAVE1
134	SAV4	SAV4
13C	SAV4	SAV4
144	SAVALL	SAVALL
14C	MATCHI	$UNRESOLVED

ENTRY ADDRESS 00
TOTAL LENGTH 11610

****MOX02DN NOW REPLACED IN DATA SET

```
(17)          OS/360  FORTRAN H                                   DATE  71.312/18.09.59

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
ISN 0002            SUBROUTINE LBONDI
                C   THIS SUBROUTINE CALCULATES THE RECURRING AND NON-RECURRING LOWER
                C     BOUND WITH A PENALTY FUNCTION INCLUDED IF W NE 1.E30
                C
ISN 0003            REAL NPERPD
ISN 0004            INTEGER*2  NSAVE,NADD,NX,MINOPT,MORE,NTGYTR,
                   1  VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,KOUT,LTR,NINTYR
                C *
ISN 0005            COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                   1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
ISN 0006            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                   1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
ISN 0007            COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
                   1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
ISN 0008            COMMON/SCRACH/EXTRA,NADD,NX,MORE(10),ZKP,WKP,NXKP,LZKP(5),DUME(11)
                   *,  A2,LZ(46),W(500),W2(500),
                   1 TDS(500),WR(499),Z(500),COST(2,250),MINOPT(246,9),NODE(5,500),
                   2  NPOS,SIGSO(9),ETC(9),
                   4  NCOST,LB,KX,KZ,NSAVE(10),KEEP(40),MZ(60),DUM
                C
ISN 0009            IF(LB.EQ.50)  GO TO 54
                C     ***FIND NEW RECURRING LOWER BOUND***
ISN 0011          49 W(KX)=0.
ISN 0012            W2(KX) = 0.0
ISN 0013            DO 50 J=1,NM
ISN 0014            IF(YRLM(J).LT..001)   GO TO 50
ISN 0016            CALL UNPACK(MZ,VNM(1,J),NV ,1)
ISN 0017            COST(1,J) = 1.0E30
ISN 0018            COST(2,J) = 1.0E30
ISN 0019            KO = LYR(J)
ISN 0020            JX = LETT(J)
ISN 0021            ITR = LTR(JX)
ISN 0022            DO 48 I1 = 1,NV
ISN 0023            IF(MZ(I1).EQ.0)  GO TO 48
ISN 0025            I = I1
ISN 0026            IF(ITR.EQ.2)  I = I1 & NV
ISN 0028            DO 47 M=1,20
ISN 0029            IF(NONREC(I,M).EQ.0) GO TO 475
ISN 0031            NO = NONREC(I,M)
ISN 0032            IF(KI*LZ(NO).LT.  KO  ) GO TO 48
ISN 0034          47 CONTINUE

ISN 0035         475 X = NMULT(I1,JX)
ISN 0036            CX=YRLM(J)*RECUR(I1,KO,ITR)*X
ISN 0037            IF(CX.GE.COST(2,J))  GO TO 48
ISN 0039            IF(CX.LT.COST(1,J))  GO TO 43
ISN 0041            COST(2,J) = CX
ISN 0042            GO TO 48
ISN 0043          43 COST(2,J) = COST(1,J)
ISN 0044            COST(1,J) = CX
ISN 0045            MIN(J) = I1
ISN 0046          48 CONTINUE
ISN 0047            W(KX)=W(KX)&COST(1,J)
ISN 0048            W2(KX) = W2(KX) & COST(2,J)
ISN 0049          50 CONTINUE
ISN 0050            IF(KX.EQ.NX)  GO TO 510
ISN 0052            KZ = KI*LZ(NCOST)
ISN 0053            IF(W(KX).LT.1.0E20)  GO TO 508
ISN 0055            TGO = 0.0
ISN 0056            GO TO 38
ISN 0057         508 IF(KZ.EQ.0)  GO TO 510
ISN 0059            KY = NSAVE(LB)
ISN 0060         512 IF(W(KX).GT.W(KY)-.0001.AND.W2(KX).GT.W2(KY)-.0001)  GO TO 38
ISN 0062            IF(W(KX).GT.W(KY)-.0001.AND.W2(KX).GT.1.0E25.AND.W2(KY)-W2(KX).LT.
                   1  1.0E25)  GO TO 38
                C
                C     CALCULATE LOWER BOUND USING PENALTY FUNCTION BASED ON VEHICLES
ISN 0064          510 DO 350 NIC = 1,NUMD
ISN 0065            KEEP(NIC) = 1
ISN 0066            IF(LZ(NIC).LT.15)  KEEP(NIC) = 0
ISN 0068          350 CONTINUE
ISN 0069          355 TGO = 0.0
ISN 0070            IV = 0
ISN 0071            TG = 0.0
ISN 0072          354 DO 351 IX = 1,NV
ISN 0073            IF(IX.EQ.IV)  GO TO 351
ISN 0075            VGO = 0.0
ISN 0076          330 DO 90 J = 1,NM
ISN 0077            IF(YRLM(J).LT..001)  GO TO 90
ISN 0079            IF(MIN(J).EQ.IX)  GO TO 91
ISN 0081           90 CONTINUE
ISN 0082            GO TO 351
ISN 0083           91 PF = 0.0
ISN 0084            KTV = 0
```

```
ISN 0085            I1 = IX
ISN 0086            JX = LETT(J)
ISN 0087            IF(LTR(JX).EQ.2)  I1 = IX & NV
ISN 0089            DO 341 M = 1,20
ISN 0090            IF(NONREC(I1,M).EQ.0)  GO TO 3415
ISN 0092            NO = NONREC(I1,M)
ISN 0093            IF(KEEP(NO).EQ.0)  GO TO 341
ISN 0095            VGO = VGO &DS(NO) & FLOAT(KI)*SUST(NO)
ISN 0096            KTV = 1
ISN 0097          341 CONTINUE
ISN 0098         3415 IF(KTV.EQ.0)  GO TO 351
ISN 0100            DO 331  J = 1,NM
ISN 0101            IF(YRLM(J).LT.0.001.OR.MIN(J).NE.IX)  GO TO 331
ISN 0103            PF = PF & COST(2,J) - COST(1,J)
ISN 0104          331 CONTINUE
ISN 0105            VGO = AMIN1(VGO,PF)
ISN 0106            IF(VGO.LT.TG)  GO TO 351
ISN 0108            IV = IX
ISN 0109            TG = VGO
ISN 0110            I2 = I1
ISN 0111          351 CONTINUE
ISN 0112            TGO = TG & TGO
ISN 0113            IF(TG.LT.GUESS*.01)  GO TO 38
ISN 0115            TG = 0.0
ISN 0116            DO 352 M = 1,20
ISN 0117            IF(NONREC(I2,M).EQ.0)  GO TO 354
ISN 0119            NO = NONREC(I2,M)
ISN 0120            KEEP(NO) = 0
ISN 0121          352 CONTINUE
ISN 0122            GO TO 354
ISN 0123           38 IF(KZ.EQ.0)  TDS(KX) = TDS(NX)
ISN 0125            IF(KZ.GT.0.AND.KX.NE.NX)  TDS (KX) = TDS(NX)
                   1           & DS(NCOST) & FLOAT(   LB*KI-NYD(NCOST)&1)*SUST(NCOST)
ISN 0127           54 IF (KX.EQ.NX) TDS(NX) = TDS(NX) &
                     1         DS(NCOST) & FLOAT(LYD(NCOST)-NYD(NCOST)&1)*SUST(NCOST)
ISN 0129            DMIN = TGO & TDS(KX)
                  C
ISN 0130          507 Z(KX) = DMIN & W(KX)
ISN 0131            IF(LP.GT.0) WRITE(6,204)  KX,NX,NCOST,KZ,W(KX),DMIN,Z(KX)
ISN 0133            RETURN
ISN 0134          204 FORMAT (1H ,4(I3,5X),3(F9.2,5X))
ISN 0135            END
```

****** END OF COMPILATION ******

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000      NAME MOX02LD(R)
IEW0461   IBCOM=
IEW0461   UNPACK

                                                CROSS REFERENCE TABLE

 CONTROL SECTION                          ENTRY
   NAME     ORIGIN   LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

 LBOUND1        00      A50
 SAV3          A50      980
 SAVALL       13D0     3A1C
 TEMP         4DF0     4110
 SCRACH       8F00     6A60


 LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      168          SAV3          SAV3                       16C          SAVALL          SAVALL
      170          SAVALL        SAVALL                     174          TEMP            TEMP
      178          SCRACH        SCRACH                     17C          SCRACH          SCRACH
      180          SCRACH        SCRACH                     184          SCRACH          SCRACH
      188          SCRACH        SCRACH                     18C          IBCOM=          $UNRESOLVED
      190          UNPACK        $UNRESOLVED                 8C          SCRACH          SCRACH
       94          SAVALL        SAVALL
 ENTRY ADDRESS     00
 TOTAL LENGTH    F960

****MOX02LD   NOW REPLACED IN DATA SET


(17)          OS/360  FORTRAN H                              DATE  71.312/17.03.49

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
   ISN 0002            SUBROUTINE LISTC
   ISN 0003            DOUBLE PRECISION NAME,NAMEN,NAMEK
   ISN 0004            INTEGER*2 KODESP,KODEM,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NYRSFX,NPROG,
                      1  KPROG,KODE,LTR,YDPL
   ISN 0005            REAL LEVEL,NPERPD
   ISN 0006            COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                      1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                      2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                      3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
                      4,  R(84), S(84),CS(90),NPROG(90),KPROG(90), KODE(90)
   ISN 0007            COMMON/SAV3/GRD,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                      1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
   ISN 0008            IF(NCSTR.EQ.0)   RETURN
   ISN 0010            WRITE (6,1)  NCSTR
   ISN 0011           1 FORMAT (1H1,25X,I2,' CONSTRAINTS'/6X,'KODE')
   ISN 0012            DO 200 I = 1,NCSTR
   ISN 0013            L = KODE(I)
   ISN 0014            IF(L.LT.1.OR.L.GT.11)   GO TO 200
   ISN 0016            J = NPROG(I)
   ISN 0017            K = KPROG(I)
   ISN 0018            IF(J.LE.NMIS & NSPR)  NAMEN = NAME(J)
   ISN 0020            NAMEK = NAME(K)
   ISN 0021            Z = CS(I)
   ISN 0022            GO TO (10,20,30,40,50,60,70,80,90,100,110),  L
   ISN 0023           10 WRITE(6,11) L,J,NAMEN,K,NAMEK,Z
   ISN 0024           11 FORMAT(6X,I3,3X,'START ',I3,1X,A6,' AFTER END ',I3,1X,A6,' &',F3.0
                        *)
   ISN 0025            GO TO 200
   ISN 0026           20 WRITE(6,21) L,J,NAMEN,Z,K,NAMEK
   ISN 0027           21 FORMAT (6X,I3,3X,'END ',I3,1X,A6,' &',F3.0,' BEFORE START ',I3,1X,
                        * A6)
   ISN 0028            GO TO 200
   ISN 0029           30 WRITE(6,31) L,J,NAMEN,Z
   ISN 0030           31 FORMAT (6X,I3,3X,'START ',I3,1X,A6,'  IN',F6.0)
   ISN 0031            GO TO 200
   ISN 0032           40 WRITE (6,41) L,J,NAMEN,Z
   ISN 0033           41 FORMAT (6X,I3,3X,'END DEVL ',I3,1X,A6,' IN',F6.0)
   ISN 0034            GO TO 200
   ISN 0035           50 WRITE (6,51) L,J,NAMEN,Z
   ISN 0036           51 FORMAT (6X,I3,3X,I3,1X,A6,F3.0,' YEARS DEVELOPMENT')
   ISN 0037            GO TO 200
   ISN 0038           60 IF (Z.LE.0.) GO TO 64
```

```
ISN 0040            WRITE (6,61) L,J,NAMEN,Z,K,NAMEK
ISN 0041         61 FORMAT (6X,I3,3X,'TARGET DATE ',I3,1X,A6,' NO LATER THAN',F4.0,
                    *  ' YEARS BEFORE ',I3,1X,A6)
ISN 0042            GO TO 200
ISN 0043         64 Z = ABS (Z)
ISN 0044            WRITE (6,65) L,J,NAMEN,Z,K,NAMEK
ISN 0045         65 FORMAT (6X,I3,3X,'TARGET DATE ',I3,1X,A6,' NO LATER THAN',F4.0,
                    *  ' YEARS AFTER  ',I3,1X,A6)
ISN 0046            GO TO 200
ISN 0047         70 WRITE (6,71) L,J,NAMEN,Z
ISN 0048         71 FORMAT (6X,I3,3X,'TARGET DATE ',I3,1X,A6,' NO LATER THAN',F6.0)
ISN 0049            GO TO 200
ISN 0050         80 IF(J.LE.NMIS & NSPR)  WRITE (6,81) L,J,NAMEN
ISN 0052         81 FORMAT (6X,I3,3X,I3,1X,A6,' FIXED')
ISN 0053            IF(J.GT.NMIS & NSPR) WRITE(6,82) L,J
ISN 0055         82 FORMAT (6X,I3,3X,'PROGRAM DEV  ', I3,'  FIXED')
ISN 0056            GO TO 200
ISN 0057         90 WRITE (6,91) L,J,NAMEN,Z
ISN 0058         91 FORMAT (6X,I3,3X,'START ',I3,1X,A6,' NO EARLIER THAN',F6.0)
ISN 0059            GO TO 200
ISN 0060        100 WRITE (6,101) L,J,NAMEN,Z
ISN 0061        101 FORMAT (6X,I3,3X,'TARGET DATE ',I3,1X,A6,' NO EARLIER THAN ',F6.0)
ISN 0062            GO TO 200
ISN 0063        110 WRITE( 6,111) L,J,K,NAMEK
ISN 0064        111 FORMAT(6X,I3,3X, 'PROGRAM DEV  ',I3, '  COMPLETED BY FIRST LAUNCH
                   1OF PROGRAM ',I3,1X,A6)
ISN 0065        200 CONTINUE
ISN 0066            RETURN
ISN 0067            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02LC(R)
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

CONTROL SECTION			ENTRY						
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME
LISTC	00	784							
SAV2	788	FE0							
SAV3	1798	980							

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
370	SAV2	SAV2	374	SAV3	SAV3
378	IBCOM=	$UNRESOLVED			

ENTRY ADDRESS 00
TOTAL LENGTH 2118

****MOX02LC NOW REPLACED IN DATA SET

```
C   OPTIMAL RESOURCE ALLOCATION MODEL
C   COST UNCERTAINTIES ARE DISPLAYED QUANTITATIVELY
C
C  A PROGRAM EITHER EQUALS A MISSION WITH LAUNCH SCHEDULE OR A
C  DEVELOPMENT OR SUSTAINING PROGRAM OR A MISC. PROGRAM
C  A MISSION MUST HAVE AT LEAST ONE AND NO MORE THAN 10 LAUNCH YEARS
C
0001        DOUBLE PRECISION NAME
0002        LOGICAL SKIP,EXT,ACCL
0003        REAL NPERPD
0004        INTEGER PROG
0005        INTEGER*2 YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,KODE,
           1  NYRSFX,KODEM,KODESP,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,
           2  MIN,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,LTR,NU,NBY,
           3        MODE,NOB,LSA,NYS,KODEF,LST,MST,IST,JST,KST,NVS,MRV,NRP,
           4  NYP,KODEP,IVEHA,NTRIP,NPLS,NRR,MR,NPSTG,NPAD,NPFAM,NFS,NPINTL,
           5  NPINTU,MAPS,MAPF,MAPI,KODEV,MAF,MAIC,
           6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
0006        COMMON/SAVER/  RFIXD(12,84)
0007        COMMON/SAVDMP/ NFAM,KFLAG,FAM(30),KODEF(30),FMNR(30),FMSUS(30),
           1JST(30),YDF(30),LSA(40),SNR(40),NYS(40),DINT(40),SINT(40),KST(40),
           2  YDI(40),YDS(40),IST(40),FMSLS(30,2),SUSLS(40,2),SINTLS(40,2),
           3  LST(30,5),YDPF(30,5),MST(30,10),YDPS(30,10)
0008        COMMON/SAVSAR/CDR,POJ(3),SRJ(3,3),NU(40),NBY(40),NOB(40),RINT(40),
           1    PLCINT(40),XLT(40), PLCT(40),UPP(40),TAT(40),TAMT(50),SR(40,3),
           2    MODE(40,3),PLC(40,3)
0009        COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
           1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
           2  RPLM(50),MAS(40,3), RXD(12,50)
0010        COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
           1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
           2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
           3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
           4,  R(84), S(84),CS(90),NPROG(90),KPROG(90), KODE(90)
0011        COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
           1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0012        COMMON/SAV4/  MAF(30,3), MAIC(40,3),
           *              NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
           1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
           2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
           3    PSTGS(30,10,2)
0013        COMMON/SVACAV/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
           1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
```



```
           2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
0014        COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
           1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
           2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0015        COMMON/VARNCE/KSTAT,VARI(40),VARF(50),VARM(56),FMVAR(2,30),
           1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)
0016        COMMON/SCRACH/M,N,NCS,PROG,IODD,IERR,SKIP,MYFLAG,JS,NSCALE(5),
           1  NSL(10),TOTAL(20),W(20),D(20),XOUT(20),VOUT(20),RRR(20),YEAR(20)
           2,  Y(20),KVEHI(50),LABEL(50),LVARY(70),LVD(70),IVEH(70),LVS(70),
           3  LVSF(80),VNAM(80),NOP(86),RF(86),CF(86),SF(86),FLAGR(86),
           4  FLAGS(86),NSSF(86),NSRF(86),NSXF(86),NDSF(86),SUSTF(86),NLVP(86)
           5,  NSTRRC(86),NYRSRC(86),LNDF(86),NSTRST(86),LNDATE(86),NPRO(90),
           6  KPRO(90),CSX(90),LZ(46),RCOST( 60), KVEH( 60),IMAGE(830),
           7  XSCH(10,70),PLSCH(10,70),XLVSUM(20,50),RECUR(20,50),KODX(90),
           8  LABN(46),DM(265)
0017        DATA BLANK /1H /
0018        STS(41) = BLANK
0019        LYR(252) = 0
0020      9 FINISH = 1
0021        KSTAT = 0
0022        JFLAG = 0
0023        DO 3 I = 1, 40
0024        DO 2 J = 1,3
0025        FIVAR(J,I) = 0.0
0026      2 SVAR(J,I) = 0.0
0027        VARI(I) = 0.0
0028        SVAR(4,I) = 0.0
0029      3 SVAR(5,I) = 0.0
0030        DO 4 I = 1,30
0031        FMVAR(1,I) = 0.0
0032      4 FMVAR(2,I) = 0.0
0033        DO 5 I = 1,56
0034        VARM(I) = 0.0
0035        DO 5 J = 1,3
0036      5 PLVAR(J,I) = 0.0
0037        DO 6 I = 1,50
0038      6 VARF(I) = 0.0
0039     10 CALL ASSIGN
0040        IF(MYRS.EQ.0.0)  GO TO 99
0041        IF(MYRS.EQ.100)  GO TO 9
0042        IF(FINISH.GT.1) GO TO 12
0043        NNM = NMIS + 1 + NSPR
0044        TREF = 1900.0 + TREF
```

```
0045              DO 8 I = 1,84
0046            8 R(I) = 0.0
0047           12 DO 13 I = 1,70
0048              LVARY(I) = 0
0049              LVD(I) = 0
0050              DO 131 J = 1,10
0051              PLSCH(J,I) = 0.0
0052          131 XSCH(J,I) = 0.0
0053           13 CONTINUE
0054              DO 132 I = 1,86
0055          132 NLVP(I) = 0
0056              DO 133 I = 1,50
0057          133 LABEL(I) = 0
0058              DO 134 I = 1,NMIS
0059          134 NYRSST(I) = 0
0060              NUMD = NUMD + NEXD
0061              DO 14 I = 1, NUMD
0062           14 LABN(I) = 0
                C
                C  CALCULATE VARIABLES FOR SMOOTH FROM MISSION DATA
                C
0063              M = 1
0064              DO 120 K = 1,NM
0065              IF(MIN(K).EQ.0)  GO TO 120
0066              I = LYR(K)
0067              J = LETT(K)
0068              IF(J.EQ.LETT(K-1))  GO TO 105
0069              IF(FINISH.GT.1)  GO TO 104
0070              S(J) = IS(J)
0071              R(J) = YDPL(J)
0072          104 LVARY(J) = M
0073              NSTRST(J) =  INT(2.0*R(J)/3.0 + .999)
0074              IF(R(J).EQ.0)  NSTRST(J)=1
0075              GO TO 108
0076          105 L1 = LVARY(J)
0077              MO = M-1
0078              DO 106 L = L1,MO
0079              IF(MIN(K).NE.IVEH(L))  GO TO 106
0080              M1 = L
0081              GO TO 110
0082          106 CONTINUE
0083          108 IVEH(M) = MIN(K)
0084              LVS(M) = I - IS(J) + 1900 + ILY
```



```
0085              NLVP(J) = NLVP(J) + 1
0086              M1 = M
0087              M = M + 1
0088          110 M3 = IS(J)
0089              IF(SUS(J).LE..001)  GO TO 111
0090              NX = NYRSST(J)
0091              M4 = NSTRST(J)
0092              M5 = NSYR(J)
0093              NYRSST(J) = MAXO(NX,I - M4 - M3 + 1900 + ILY + M5 + 1)
0094          111 M2 = LVS(M1)
0095            * K1 = MIN(K)
0096              X = NMULT(K1,J)
0097              NDUM = I - M2 - M3 + 1900 + ILY + 1
0098              PLSCH(NDUM,M1) = YRLM(K)
0099              XSCH(NDUM,M1) = YRLM(K)*X
0100              NX = LVD(M1)
0101              LVD(M1) = MAXO(NX,NDUM)
0102          120 CONTINUE
0103              M = M - 1
0104              NCS = 0
0105              N = NMIS
0106              IF(NSPR.EQ.0)  GO TO 170
0107              DO 150 I = 1,NSPR
0108              N = N + 1
0109              IF(FINISH.GT.1)  GO TO  140
0110              S(N) = IS(N)
0111              R(N) = YDPL(N)
0112          140 NSTRST(N) = INT(2.0*R(N)/3.0 + .999)
0113              IF(R(N).EQ.0)  NSTRST(N) = 1
0114          150 CONTINUE
                C
                C  CONTINUE TO CALCULATE VARIABLES FOR SMOOTH USING DEV. AND SUST. COSTS
                C
0115          170 IF(NUMD.EQ.0)  GO TO 260
0116              CALL UNPACK(LZ,LZOPT(1),NUMD,5)
0117              DO 210 I = 1,NUMD
0118              IF(LZ(I).EQ.0)  GO TO 210
0119              N = N + 1
0120              NDUM = N - NMIS - NSPR
0121              LABEL(NDUM) = I
0122              LABN(I) = N
0123              C(N) = DS(I)
0124              NYRSFX(N) = 0
```

```
0125              L = MAT(I)
0126              IF(L.GT.1000)  GO TO 206
0127              IF(L.LT.-100)  NDUM = -L -100
0128              IF(L.LT.-100)  J = LABI(NDUM)
0129              IF(L.LT.0.AND.L.GE.-100)  NDUM = -L
0130              IF(L.LT.0.AND.L.GE.-100)  J = LABF(NDUM)
0131              IF(L.GT.0)  J = LABS(L)
0132              IF(J.EQ.0)  GO TO 206
0133              DO 205  K = 1,12
0134              RFIXD(K,N) = RXD(K,J)
0135          205 C(N) = C(N) - RXD(K,J)
0136              NYRSFX(N) = NSFX(J)
0137              NSTRFX(N) = NRFX(J)
0138          206  NDUM = I + NMIS + NSPR
0139              S(N) = IS(NDUM)
0140              R(N) = YD(I)
0141              SUS(N) = SUST(I)
0142              NSTRST(N) = INT(2.0*R(N)/3.0 + .999)
0143              IF(R(N).EQ.0)  NSTRST(N)=1
0144              NX = IS(NDUM) + NSTRST(N) - 1
0145              NX = MAXO (0,1900 + ILY - NX)
0146              NYRSST(N) = LZ(I) - NYD(I) + INT(YD(I)) - NSTRST(N) + 2 + NX
0147              IF(SUS(N).LT..0001)  NYRSST(N) = 0
0148          210 CONTINUE
                C
                C  CALCULATE DEVELOPMENT CONSTRAINTS ON MISSION PROGRAMS
0149              DO 250  K = 1,NM
0150              IF(MIN(K).EQ.0)   GO TO 250
0151              J = LETT(K)
0152              IF(NLVP(J).EQ.1.AND.J.EQ.LETT(K-1))  GO TO 250
0153              IV = MIN(K)
0154              DO 211 I = 1,10
0155              NDM = K - I
0156              IF(J.NE.LETT(NDM)) GO TO 215
0157              IF (IV.EQ.MIN(NDM)) GO TO 250
0158          211 CONTINUE
0159          215 X = LYR(K) - LYR(NDM+1)
0160              I1 = IV
0161              IF(LTR(J).EQ.2)  I1 = IV + NV
0162              DO 220 K1 = 1,20
0163              IF(NONREC(I1,K1).EQ.0)  GO TO 250
0164              NO = NONREC(I1,K1)
0165              J1 = LABN(NO)
```



```
0166              NCS = NCS + 1
0167              NPRO(NCS) = J1
0168              CSX(NCS) = -1.0 -X
0169              IF(C(J1).LT..0001)  GO TO 216
0170              KPRO (NCS) = J
0171              KODX(NCS) = 11
0172              GO TO 217
0173          216 IF(NCS.EQ.1)  GO TO 219
0174              NCS = NCS - 1
0175              DO 218 I = 1,NCS
0176              IF(J1.EQ.NPRO(I))  GO TO 220
0177          218 CONTINUE
0178              NCS = NCS + 1
0179          219 KPRO(NCS) = 0
0180              KODX(NCS) = 8
0181          217  IF(NCS.GE.90)  GO TO 255
0182          220 CONTINUE
0183          250 CONTINUE
0184              GO TO 260
0185          255 WRITE(6,1002)
0186         1002 FORMAT(52HONUMBER OF DEVELOPMENT CONSTRAINTS HAS BEEN EXCEEDED)
                C
0187          260 CALL SMOOTH
                C
0188              IF(MOS.EQ.2.OR.MOS.EQ.3)  GO TO 9
0189              IF(FINISH.EQ.MITR + 1.AND.JFLAG.EQ.1)  GO TO 401
0190              IF(FINISH.EQ.MITR + 1)  GO TO 402
0191              IF(FINISH.EQ.MITR)   JFLAG = 1
                C
                C  CALCULATE VARIABLES FOR ASSIGN FROM SMOOTH VARIABLES
                C
0192              MXRS = MYRS
0193              DO 300  K = 1,NM
0194              I = LYR(K)
0195              J = LETT(K)
0196              IF(J.EQ.LETT(K-1))  GO TO 305
0197              IS(J) = S(J)
0198              IX = IS(J) + LNDATE(J) - 1900 - ILY
0199              IDIFF = IX - I
0200          305 IF (IDIFF.EQ.0)  GO TO 300
0201              MYRS = MAXO(MYRS,IDIFF + I)
0202              LYR(K) = I + IDIFF
0203          300 CONTINUE
```

```
FORTRAN IV G LEVEL 1, MOD 4          MASTER          DATE = 71312          17/27/34

0204              MYRS = MINO (MYRS,20)
0205              IF(N.EQ.NMIS+NSPR)  GO TO 10
0206              DO 350 I = NNM , N
0207              NDUM = I - NMIS - NSPR
0208              J = LABEL(NDUM)
0209              DS(J) = C(I)
0210              L = MAT(J)
0211             *IF(L.GT.1000)  GO TO 320
0212              IF(L.LT.-100)  NDM = -L -100
0213              IF(L.LT.-100)  J1 = LABI(NDM)
0214              IF(L.LT.0.AND.L.GE.-100)  NDM = -L
0215              IF(L.LT.0.AND.L.GE.-100)  J1 = LABF(NDM)
0216              IF(L.GT.0)  J1 = LABS(L)
0217              IF(J1.EQ.0)  GO TO 320
0218              DO 310 K = 1,12
0219          310 DS(J) = DS(J) + RFIXD(K,I)
0220              NREX(J1) = NSTREX(I)
0221          320 SUST(J) = SUS(I)
0222              YD(J) = R(I)
0223              NYD(J) = INT(S(I) + R(I)) - 1900 - ILY
0224              IF(NYD(J).LE.0)   NYD(J) = 1
0225              NDUM = J + NMIS + NSPR
0226              IS(NDUM) = S(I)
0227          350 CONTINUE
0228              NUMD = NUMD - NEXD
0229              DO 349 I = 1,NUMD
0230              IF(LYD(I).EQ.MXRS)   LYD(I) = MYRS
0231          349 CONTINUE
0232              GO TO 10
0233          401 WRITE(6,500)
0234              GO TO 9
0235          402 WRITE(6,501)
0236              GO TO 9
0237          500 FORMAT (58HOMAXIMUM NUMBER OF ITERATIONS COMPLETED - END OF THIS C
                 *ASE)
0238          501 FORMAT ( 65HOOPTIMUM ASSIGNMENT WITHIN BUDGET CONSTRAINTS HAS BEEN
                 1 DETERMINED)
0239           99 STOP
0240              END

FORTRAN IV G LEVEL 1, MOD 4          MASTER          DATE = 71312          17/27/34

  TOTAL MEMORY REQUIREMENTS 001984 BYTES
```

```
F83-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000      NAME MOX02MN(R)
IEW0461  ASSIGN
IEW0461  UNPACK
IEW0461  IBCOM=
IEW0461  SMOOTH
IEW0461  MAXO
IEW0461  MINO
```

MODULE MAP

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME
MASTER	00	1984							
SAVER	1988	FC0							
SAVDMP	2948	148C							
SAVSAR	3E08	A5C							
SAVE1	4868	FC4							
SAV2	5830	FE0							
SAV3	6810	980							
SAV4	7190	3188							
SVACAV	A318	848							
SAVALL	AE60	3A1C							
VARNCE	E880	ADC							
SCRACH	F360	6A60							

ENTRY ADDRESS 00
TOTAL LENGTH 15DC0

****MOX02MN NOW REPLACED IN DATA SET

FORTRAN IV G LEVEL 1, MOD 4 MATCHI DATE = 71312 16/54/25

```
0001                  SUBROUTINE MATCHI
                C     ***MATCH DECISION COSTS WITH EACH VEHICLE***
                C
0002                  INTEGER*2  LSA,NYS,KODEF,LST,MST,IST,JST,KST,VEH,NMULT,NONREC,NYD,
                     1  IS,MAT,LYR,LETT,LYD,MIN,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,MAPS,
                     2  MAPF,MAPI,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,MAIC,MAF
                C
0003                  COMMON/SAVDMP/ NFAM,KFLAG,FAM(30),KODEF(30),FMNR(30),FMSUS(30),
                     1JST(30),YDF(30),LSA(40),SNR(40),NYS(40),DINT(40),SINT(40),KST(40),
                     2  YDI(40),YDS(40),IST(40),FMSLS(30,2),SUSLS(40,2),SINTLS(40,2),
                     3  LST(30,5),YDPF(30,5),MST(30,10),YDPS(30,10)
0004                  COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                     1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                     2  RPLM(50),MAS(40,3), RXD(12,50)
0005                  COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                     1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0006                  COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                     *             NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                     1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                     2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                     3   PSTGS(30,10,2)
0007                  COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMO,MYRS,LZOPT(8),NYD(46),MAT(46
                     1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                     2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
                C
0008                  IF(LP.GE.2)  WRITE(6,251)
0009                4 DO 66 I = 1,NV
0010                  I2 = I + NV
0011                  JX = 0
0012                  KX = 0
0013                  DO 64 J = 1,20
0014                  NONREC(I2,J) = 0
0015               64 NONREC(I,J) = 0
0016               25 DO 65 MS= 1,4
0017                  K = VEH(MS,I)
0018                  IF(K.EQ.0) GO TO 66
0019                  IF (MAS(K,1) .EQ. 0) GO TO 9050
0020                  JX = JX + 1
0021                  NONREC(I,JX) = MAS(K,1)
0022                  KX = KX + 1
0023                  NONREC(I2  ,KX) = MAS(K,1)
0024                  IF(LP.GE.2)  WRITE(6,250)  I,  MAS(K,1)
0025                  IF(LP.GE.2)  WRITE(6,250) I2, MAS(K,1)
```

```
0026              IF(JX.GT.20.OR.KX.GT.20)  GO TO 93
0027           9050 IF(NP.EQ.0)  GO TO 63
0028              IF (MAS(K,2).EQ.0) GO TO 9051
0029              DO 302 L = 1,JX
0030              IF(MAS(K,2).EQ.NONREC(I,L))   GO TO 9051
0031            302 CONTINUE
0032              JX = JX + 1
0033              NONREC(I,JX) = MAS(K,2)
0034              IF(LP.GE.2)  WRITE(6,250)  I,  MAS(K,2)
0035              IF (JX.GT.20) GO TO 93
0036           9051 IF (MAS(K,3).EQ.0) GO TO 9052
0037              DO 303 L = 1,KX
0038              IF(MAS(K,3).EQ.NONREC(I2,L))   GO TO 9052
0039            303 CONTINUE
0040              KX = KX + 1
0041              NONREC(I2  ,KX) = MAS(K,3)
0042              IF(LP.GE.2)  WRITE(6,250)  I2, MAS(K,3)
0043              IF (KX.GT.20) GO TO 93
0044           9052 IF (NPAD(1,I).EQ.0) GO TO 9053
0045              N1 = NPAD(1,I)
0046              DO 9054 J = 1,10
0047              IF (NPSTG(N1,J) .NE.K) GO TO 9054
0048              IF (MAPS(N1,J).EQ.0) GO TO 9053
0049              DO 304 L = 1,JX
0050              IF(MAPS(N1,J).EQ.NONREC(I,L))   GO TO 9053
0051            304 CONTINUE
0052              JX = JX + 1
0053              NONREC(I,JX) = MAPS(N1,J)
0054              IF(LP.GE.2)  WRITE(6,250)  I,  MAPS(N1,J)
0055              IF (JX.GT.20) GO TO 93
0056              GO TO 9053
0057           9054 CONTINUE
0058           9053 IF (NPAD(2,I).EQ.0) GO TO 63
0059              N1 = NPAD(2,I)
0060              DO 9055 J = 1,10
0061              IF (NPSTG(N1,J).NE.K) GO TO 9055
0062              IF (MAPS(N1,J).EQ.0) GO TO 63
0063              DO 305 L = 1,KX
0064              IF(MAPS(N1,J).EQ.NONREC(I2,L))   GO TO 63
0065            305 CONTINUE
0066              KX = KX + 1
0067              NONREC(I2  ,KX) = MAPS(N1,J)
0068              IF(LP.GE.2)  WRITE(6,250)  I2, MAPS(N1,J)
```



```
0069              IF (KX.GT.20) GO TO 93
0070              GO TO 63
0071           9055 CONTINUE
               C      ***   PICK UP SHARED COSTS   ***
0072              63 IF(NFAM.EQ.0)  GO TO 21
0073              DO 885 KY=1,4
0074              KZ=NFS(K,KY)
0075              IF(KZ.EQ.0) GO TO 885
0076              IF (MAF(KZ,1).EQ.0) GO TO 9056
0077             *DO 306 L = 1,JX
0078              IF(MAF(KZ,1).EQ.NONREC(I,L))  GO TO 401
0079            306 CONTINUE
0080              JX=JX+1
0081              NONREC(I,JX) = MAF(KZ,1)
0082              IF(LP.GE.2)  WRITE(6,250)  I,  MAF(KZ,1)
0083            401 DO 307 L = 1,KX
0084              IF(MAF(KZ,1).EQ.NONREC(I2,L))   GO TO 9056
0085            307 CONTINUE
0086              KX = KX + 1
0087              NONREC(I2  ,KX) = MAF(KZ,1)
0088              IF(LP.GE.2)  WRITE(6,250)  I2, MAF(KZ,1)
0089              IF(JX.GT.20.OR.KX.GT.20)    GO TO 93
0090           9056 IF(NP.EQ.0)  GO TO 885
0091              IF (MAF(KZ,2).EQ.0) GO TO 9057
0092              DO 308 L = 1,JX
0093              IF(MAF(KZ,2).EQ.NONREC(I,L))    GO TO 9057
0094            308 CONTINUE
0095              JX = JX + 1
0096              NONREC(I,JX) = MAF(KZ,2)
0097              IF(LP.GE.2)  WRITE(6,250)  I,  MAF(KZ,2)
0098              IF(JX.GT.20) GO TO 93
0099           9057 IF (MAF(KZ,3).EQ.0) GO TO 9058
0100              DO 309 L = 1,KX
0101              IF(MAF(KZ,3).EQ.NONREC(I2,L))   GO TO 9058
0102            309 CONTINUE
0103              KX = KX + 1
0104              NONREC(I2  ,KX) = MAF(KZ,3)
0105              IF(LP.GE.2)  WRITE(6,250)  I2, MAF(KZ,3)
0106              IF (KX.GT.20) GO TO 93
0107           9058 IF (NPAD(1,I).EQ.0) GO TO 9059
0108              N1 = NPAD(1,I)
0109              DO 9060 J = 1,5
0110              IF (NPFAM(N1,J).NE.KZ) GO TO 9060
```

```
0111              IF (MAPF(N1,J).EQ.0) GO TO 9059
0112              DO 310 L = 1,JX
0113              IF(MAPF(N1,J).EQ.NONREC(I,L))  GO TO 9059
0114          310 CONTINUE
0115              JX = JX + 1
0116              NONREC(I,JX) = MAPF(N1,J)
0117              IF(LP.GE.2)  WRITE(6,250)  I,  MAPF(N1,J)
0118              IF (JX.GT.20) GO TO 93
0119              GO TO 9059
0120         9060 CONTINUE
0121         9059 IF (NPAD(2,I).EQ.0) GO TO 885
0122              N1 = NPAD(2,I)
0123              DO 9061 J = 1,5
0124              IF (NPFAM(N1,J).NE.KZ) GO TO 9061
0125              IF (MAPF(N1,J).EQ.0) GO TO 885
0126              DO 311 L = 1,KX
0127              IF(MAPF(N1,J).EQ.NONREC(I2,L))   GO TO 885
0128          311 CONTINUE
0129              KX = KX + 1
0130              NONREC(I2  ,KX) = MAPF(N1,J)
0131              IF(LP.GE.2)  WRITE(6,250)  I2, MAPF(N1,J)
0132              IF (KX.GT.20) GO TO 93
0133              GO TO 885
0134         9061 CONTINUE
0135          885 CONTINUE
                C     ***   PICK UP INTEGRATION COSTS   ***
0136           21 IF(MS.EQ.4) GO TO 65
0137              IF(VEH(MS+1,I).EQ.0) GO TO 65
0138              K1=VEH(MS+1,I)
0139              IF(NCI.EQ.0) GO TO 9062
0140              DO 89 J=1,NCI
0141              DO 887 KY=1,4
0142              IF(NFML(J).NE.NFS(K,KY)) GO TO 887
0143              DO 886 KZ=1,4
0144              IF(NFMU(J).EQ.NFS(K1,KZ)) GO TO 888
0145          886 CONTINUE
0146          887 CONTINUE
0147              GO TO 89
0148          888 IF (MAIC(J,1).EQ.0) GO TO 9063
0149              DO 312 L = 1,JX
0150              IF(MAIC(J,1).EQ.NONREC(I,L))   GO TO 402
0151          312 CONTINUE
0152              JX = JX + 1
```



```
0153              NONREC(I,JX) = MAIC(J,1)
0154              IF(LP.GE.2)  WRITE(6,250)  I,  MAIC(J,1)
0155          402 DO 313 L = 1,KX
0156              IF(MAIC(J,1).EQ.NONREC(I2,L))  GO TO 9063
0157          313 CONTINUE
0158              KX = KX + 1
0159              NONREC(I2  ,KX) = MAIC(J,1)
0160              IF(LP.GE.2)  WRITE(6,250)  I2, MAIC(J,1)
0161              IF(JX.GT.20.OR.KX.GT.20)    GO TO 93
0162         9063 IF(NP.EQ.0)   GO TO 89
0163              IF (MAIC(J,2).EQ.0) GO TO 9064
0164              DO 314 L = 1,JX
0165              IF(MAIC(J,2).EQ.NONREC(I,L))  GO TO 9064
0166          314 CONTINUE
0167              JX = JX + 1
0168              NONREC(I,JX) = MAIC(J,2)
0169              IF(LP.GE.2)  WRITE(6,250)  I,  MAIC(J,2)
0170              IF(JX.GT.20) GO TO 93
0171         9064 IF (MAIC(J,3).EQ.0) GO TO 89
0172              DO 315 L = 1,KX
0173              IF(MAIC(J,3).EQ.NONREC(I2,L))   GO TO 89
0174          315 CONTINUE
0175              KX = KX + 1
0176              NONREC(I2  ,KX) = MAIC(J,3)
0177              IF(LP.GE.2)  WRITE(6,250)  I2, MAIC(J,3)
0178              IF (KX.GT.20) GO TO 93
0179           89 CONTINUE
0180         9062 IF(NP.EQ.0)  GO TO 65
0181              DO 9065 M = 1,2
0182              IF (NPAD(M,I).EQ.0) GO TO 9065
0183              N1 = NPAD(M,I)
0184              DO 9066 J = 1,4
0185              IF (NFS(K,J).EQ.0) GO TO 9065
0186              DO 9067 KY = 1,5
0187              IF(NPINTL(N1,KY).EQ.0)   GO TO 9066
0188              IF (NPINTL(N1,KY).NE.NFS(K,J)) GO TO 9067
0189              DO 9068 KZ = 1,4
0190              IF (NPINTU(N1,KY).EQ.NFS(K1,KZ)) GO TO 9069
0191         9068 CONTINUE
0192              GO TO 9067
0193         9069 IF (MAPI(N1,KY).EQ.0) GO TO 9067
0194              IF (M.EQ.2) GO TO 9070
0195              DO 316 L = 1,JX
```

```
FORTRAN IV G LEVEL 1, MOD 4            MATCHI            DATE = 71312          16/54/25

0196                IF(MAPI(N1,KY).EQ.NONREC(I,L))  GO TO 9067
0197              316 CONTINUE
0198                  JX = JX + 1
0199                  NONREC(I,JX) = MAPI(N1,KY)
0200                  IF(LP.GE.2)  WRITE(6,250)  I,  MAPI(N1,KY)
0201                  IF (JX.GT.20) GO TO 93
0202                  GO TO 9067
0203             9070 DO 317  L = 1,KX
0204                  IF(MAPI(N1,KY).EQ.NONREC(I2,L))  GO TO 9067
0205              317 CONTINUE
0206                  KX = KX + 1
0207                  NONREC(I2  ,KX) = MAPI(N1,KY)
0208                  IF(LP.GE.2)  WRITE(6,250)  I2, MAPI(N1,KY)
0209                  IF (KX.GT.20) GO TO 93
0210             9067 CONTINUE
0211             9066 CONTINUE
0212             9065 CONTINUE
0213               65 CONTINUE
0214               66 CONTINUE
0215                  RETURN
0216               93 WRITE(6,220) I
0217                  KFLAG = 1
0218               99 RETURN
0219              220 FORMAT(45HOEXCEEDED 20  NON-RECURRING COSTS FOR VEHICLE,I4)
0220              250 FORMAT(I5, 16X,I4)
0221              251 FORMAT (8H1VEHICLE,10X,15HDECISION NUMBER)
0222                  END


                    TOTAL MEMORY REQUIREMENTS 001984 BYTES


 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
           VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000      NAME MOX02MH(R)
IEW0461   IBCOM=


                                        MODULE MAP

  CONTROL SECTION                    ENTRY

    NAME     ORIGIN   LENGTH          NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

  MATCHI         00     1984
  SAVDMP       1988     14BC
  SAVE1        2E48      FC4
  SAV3         3E10      9E4
  SAV4         47F8     3188
  SAVALL       7980     3A1C


 ENTRY ADDRESS       00
 TOTAL LENGTH      83A0

****MOX02MH   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                DATE  71.312/18.06.54

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
ISN 0002            SUBROUTINE MATEI
                  C  DETERMINE IF VARIOUS STAGE COMBINATIONS MAKE A FEASIBLE VEHICLE
ISN 0003            REAL ISP,ISPA,LENT
ISN 0004            INTEGER*2 FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,VEH,NYD,
                   1  NMULT,NONREC,IS,MAT,LYR,LETT,LYD,MIN,NVS,MRV,NRP,NYP,KODEP,
                   2  IVEHA,NTRIP,NPLS,NRR,MR,KODEV
                  C
ISN 0005            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                   1J,SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
ISN 0006            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                   1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                   2  RPLM(50),MAS(40,3), RXD(12,50)
ISN 0007            COMMON/SVACAT/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
                   1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
                   2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
ISN 0008            COMMON/SCRACH/IP,IV,IG,NPAX(2),NEH(4),NST(41),THRT(41),DIAM(41),
                   1  TSL(41),LENT(41),WTFU(41),WTIN(41),ISP(41),MZ(50),LZ(50),
                   2          WINT(3,60),KX,NX,WGHT(40),WF(4),WT(4),ISPA(4),
                   3  THUT(4),PRT(60),M,VDES,WPL,PR,K1,IERR,DUM(6067)
                  C
ISN 0009            DATA PI,N,VREF,CLI/3.1416,2,25573.,28.5/
ISN 0010            NV1 = NV & 1
ISN 0011            DO 34 I = NV1,60
ISN 0012            VEH(1,I)=0
ISN 0013            DO 34 J=1,3
ISN 0014            VEH(J&1,I)=0
ISN 0015           34 WINT(J,I)=0.0
ISN 0016            NX = 0
ISN 0017            IMAX=0
ISN 0018            JMAX=0
ISN 0019            KMAX=0
ISN 0020            LMAX=0
ISN 0021            DO 35 I=1,NSTG
ISN 0022            IF(NST(I).EQ.0)   GO TO 36
ISN 0024           33 WGHT(I)=WTFU(I)&WTIN(I)
ISN 0025            IF(NST(I).EQ.1) IMAX=I
ISN 0027            IF(NST(I).EQ.2) JMAX=I
ISN 0029            IF(NST(I).EQ.3) KMAX=I
ISN 0031            IF(NST(I).EQ.4) LMAX=I
ISN 0033           35 CONTINUE
ISN 0034           36 IF(IMAX.EQ.0) GO TO 600
ISN 0036            IF(JMAX.EQ.0) JMAX=IMAX
ISN 0038            IF(KMAX.EQ.0) KMAX=JMAX
ISN 0040            IF(LMAX.EQ.0) LMAX=KMAX
ISN 0042            IM1=IMAX&1
ISN 0043            JM1=JMAX&1
ISN 0044            DO 500 I=1,IMAX
ISN 0045            K1 = I
ISN 0046            WF(1)=WTFU(I)
ISN 0047            WT(1)=WTIN(I)
ISN 0048            ISPA(1)=ISP(I)
ISN 0049            THUT(1)=THRT(I)
ISN 0050            DO 400 J=IM1,KMAX
ISN 0051            WINX=PI*(DIAM(I)&DIAM(J))*(SQRT((LENT(J))**2&((DIAM(I)-DIAM(J))*
                   1  0.5)**2))*5.0*0.5
ISN 0052            IF (THRT(I).LT.1.2*(WGHT(I)&WGHT(J)&WINX)) GO TO 400
ISN 0054            IF (THRT(I).GT.3.5*(WGHT(I)&WGHT(J)&WINX)) GO TO 400
ISN 0056            IF (DIAM(J).GT.1.2*DIAM(I)) GO TO 400
ISN 0058            IF (DIAM(I).GT.3.5*DIAM(J)) GO TO 400
ISN 0060            WF(2)=WTFU(J)
ISN 0061            WT(2)=WTIN(J)
ISN 0062            ISPA(2)=ISP(J)
ISN 0063            THUT(2)=THRT(J)
ISN 0064            M=0
ISN 0065            PR=0.
ISN 0066            VDES=0.
ISN 0067            CALL PERFI(CLI,N,VREF)
ISN 0068            IF (IERR.NE.0) GO TO 60
ISN 0070            NX = NX & 1
ISN 0071            KX = NV & NX
ISN 0072            NJ=J
ISN 0073            VEH(1,KX)=I
ISN 0074            VEH(2,KX)=J
ISN 0075            WINT(1,KX)=WINX
ISN 0076            PRT(KX)=WPL
                  C
ISN 0077            CALL MISMTI
                  C
ISN 0078            IF(KX.EQ.100)  GO TO 60
ISN 0080            IF(KX.GE.60)  GO TO 600
ISN 0082           60 DO 300 K=IM1,LMAX
ISN 0083            WINY=PI*(DIAM(J)&DIAM(K))*(SQRT((LENT(K))**2&((DIAM(J)-DIAM(K))
                   1  *0.5)**2))*5.0*0.5
```

```
ISN 0084            IF (THRT(J).LT.0.37*(WGHT(J)&WGHT(K)&WINY)) GO TO 300
ISN 0086            IF (THRT(J).GT.1.25*(WGHT(J)&WGHT(K)&WINY)) GO TO 300
ISN 0088            IF (THRT(I).LT.1.2*(WGHT(I)&WGHT(J)&WGHT(K)&WINX&WINY)) GO TO 300
ISN 0090            IF (THRT(I).GT.3.0*(WGHT(I)&WGHT(J)&WGHT(K)&WINX&WINY)) GO TO 300
ISN 0092            IF (DIAM(K).GT.1.2*DIAM(J)) GO TO 300
ISN 0094            IF (DIAM(J).GT.3.5*DIAM(K)) GO TO 300
ISN 0096            WF(3)=WTFU(K)
ISN 0097            WT(3)=WTIN(K)
ISN 0098            ISPA(3)=ISP(K)
ISN 0099            THUT(3)=THRT(K)
ISN 0100            M=1
ISN 0101            PR=0.
ISN 0102            VDES=0.
ISN 0103            CALL PERFI(CLI,N,VREF)
ISN 0104            IF (IERR.NE.0) GO TO 70
ISN 0106            NX = NX & 1
ISN 0107            KX = NV & NX
ISN 0108            VEH(1,KX)=I
ISN 0109            VEH(2,KX)=J
ISN 0110            VEH(3,KX)=K
ISN 0111            WINT(1,KX)=WINX
ISN 0112            WINT(2,KX)=WINY
ISN 0113            PRT(KX)=WPL
                C
ISN 0114            CALL MISMTI
                C
ISN 0115            IF(KX.EQ.100)  GO TO 69
ISN 0117            IF(KX.GE.60)  GO TO 600
ISN 0119         69 IF(K.GT.KMAX) GO TO 300
ISN 0121         70 DO 200 L=JM1,LMAX
ISN 0122            IF(L.EQ.NJ) GO TO  200
ISN 0124            WINZ=PI*(DIAM(K)&DIAM(L))*(SQRT((LENT(L))**2&((DIAM(K)-DIAM(L))
                   1  *0.5)**2))*5.0*0.5
ISN 0125            IF(THRT(K).LT.0.30*(WGHT(K)&WGHT(L)&WINZ)) GO TO 200
ISN 0127            IF(THRT(K).GT.1.25*(WGHT(K)&WGHT(L)&WINZ)) GO TO 200
ISN 0129            IF(THRT(J).LT.0.32*(WGHT(J)&WGHT(K)&WGHT(L)&WINY&WINZ)) GO TO 200
ISN 0131            IF(THRT(J).GT.1.50*(WGHT(J)&WGHT(K)&WGHT(L)&WINY&WINZ)) GO TO 200
ISN 0133            IF(THRT(I).LT.1.20*(WGHT(I)&WGHT(J)&WGHT(K)&WGHT(L)&WINX&WINY&WINZ
                   1  )) GO TO 200
ISN 0135            IF(THRT(I).GT.3.00*(WGHT(I)&WGHT(J)&WGHT(K)&WGHT(L)&WINX&WINY&WINZ
                   1  )) GO TO 200
ISN 0137            IF(DIAM(L).GT.1.2*DIAM(K)) GO TO 200


ISN 0139            IF(DIAM(K).GT.4.0*DIAM(L)) GO TO 200
ISN 0141            WF(4)=WTFU(L)
ISN 0142            WT(4)=WTIN(L)
ISN 0143            ISPA(4)=ISP(L)
ISN 0144            THUT(4)=THRT(L)
ISN 0145            M=2
ISN 0146            PR=0.
ISN 0147            VDES=0.
ISN 0148            CALL PERFI(CLI,N,VREF)
ISN 0149            IF(IERR.NE.0) GO TO 200
ISN 0151            NX = NX & 1
ISN 0152            KX = NV & NX
ISN 0153            VEH(1,KX)=I
ISN 0154            VEH(2,KX)=J
ISN 0155            VEH(3,KX)=K
ISN 0156            VEH(4,KX)=L
ISN 0157            WINT(1,KX)=WINX
ISN 0158            WINT(2,KX)=WINY
ISN 0159            WINT(3,KX)=WINZ
ISN 0160            PRT(KX)=WPL
                C
ISN 0161            CALL MISMTI
                C
ISN 0162            IF(KX.EQ.100)  GO TO 200
ISN 0164            IF(KX.GE.60)  GO TO 600
ISN 0166        200 CONTINUE
ISN 0167        300 CONTINUE
ISN 0168        400 CONTINUE
ISN 0169        500 CONTINUE
ISN 0170        600 NV = NV & NX
ISN 0171            RETURN
ISN 0172            END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000       NAME MOX02ME(R)
IEW0461  PERFI
IEW0461  SQRT
IEW0461  MISMTI

                                                CROSS REFERENCE TABLE

  CONTROL SECTION                          ENTRY

    NAME     ORIGIN  LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

  MATEI          00     A68
  SAVALL        A68    3A1C
  SAVE1        4488     FC4
  SVACAV       5450     B48
  SCRACH       5F98    6A60


   LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

       140           SAVALL         SAVALL                    144           SAVALL         SAVALL
       148           SAVE1          SAVE1                     14C           SVACAV         SVACAV
       150           SCRACH         SCRACH                    154           PERFI          $UNRESOLVED
       158           SQRT           $UNRESOLVED               15C           MISMTI         $UNRESOLVED

  ENTRY ADDRESS       00
  TOTAL LENGTH      C9F8

****MOX02ME   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                  DATE  71.312/17.04.26

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
  ISN 0002              SUBROUTINE MEAN (P1,KSTAT,SIGS,SX,SY)
                     C  FROM MODAL VALUE AND X PERCENT TAIL VALUE, CALCULATE MEAN AND SIGMA-SQUARE
                     C
  ISN 0003              KSTAT = 1
  ISN 0004              P = 1.0 - P1
  ISN 0005              CALL NDTRI(P,Y,C,IE)
  ISN 0006              SIGS = -.5*Y & .5*SQRT(Y*Y & 4.0*ALOG(2.0*SX/SY - 1.0))
  ISN 0007              SIGS = SIGS*SIGS
  ISN 0008              SY = .5*SY*(EXP(1.5*SIGS) & 1.0)
  ISN 0009              RETURN
  ISN 0010              END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000       NAME MOXO2EX(R)
IEW0461  NDTRI
IEW0461  EXP
IEW0461  SQRT
IEW0461  ALOG

                                                  CROSS REFERENCE TABLE

  CONTROL SECTION                          ENTRY
    NAME     ORIGIN   LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME
  MEAN           00      23A

  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
       00           NDTRI          $UNRESOLVED                D4           EXP            $UNRESOLVED
       D8           SQRT           $UNRESOLVED                DC           ALOG           $UNRESOLVED

 ENTRY ADDRESS       00
 TOTAL LENGTH       240

****MOXO2EX   NOW REPLACED IN DATA SET

(17)          OS/360  FORTRAN H                                  DATE  71.312/18.05.47

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
 ISN 0002            SUBROUTINE MISMTI
                   C  DETERMINE PERFORMANCE OF NEW VEHICLE IN TERMS OF MISSION MODEL INPUT
 ISN 0003            REAL LENT,ISP,ISPA,NPERPD
 ISN 0004            INTEGER*2 VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,LTR,NVS,
                    1  MRV,NRP,NYP,KODEP,IVEHA,NTRIP,NPLS,NRR,MR,KODEV,FINISH,NSTG,
                    2  NFML,NFMU,KODS,MAS,LABS,LABF,LABI
                   C
 ISN 0005            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                    1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                    2  RPLM(50),MAS(40,3), RXD(12,50)
 ISN 0006            COMMON/SVACAV/  KNV,NOPT,KODEP(30),RPLO(40),IVEHA(50),NTRIP(50),
                    1  NPLS(50),NRR(50),MR(50),NVS(60),MRV(60),NRP(60),B1(60),B2(60),
                    2  B3(60),B4(60),KODEV(60),NYP(2,60),VM(2,60)
 ISN 0007            COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                    1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
 ISN 0008            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                    1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                    2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
 ISN 0009            COMMON/SCRACH/IP,IV,IG,NPAX(2),NEH(4),NST(41),THRT(41),DIAM(41),
                    1  TSL(41),LENT(41),WTFU(41),WTIN(41),ISP(41),MZ(50),LZ(50),
                    2            WINT(3,60),KX,NX,WGHT(40),WF(4),WT(4),ISPA(4),
                    3  THUT(4),PRT(60),M,VDES,WPL,PR,K1,IERR,DUM(6067)
 ISN 0010            DATA VREF,CL1/25573.,28.5/
                   C
 ISN 0011            CF1=0.
 ISN 0012            KNS = 1
 ISN 0013            IF(NV.EQ.0)  GO TO 101
 ISN 0015            DO 100 MJ=1,NV
 ISN 0016            IF(VEH(1,KX).NE.VEH(1,MJ))  GO TO 100
 ISN 0018            IF(VEH(2,KX).NE.VEH(2,MJ))  GO TO 100
 ISN 0020            IF(VEH(3,KX).NE.VEH(3,MJ))  GO TO 20
 ISN 0022            IF(VEH(4,KX).NE.VEH(4,MJ))  GO TO 30
 ISN 0024            KX = 100
 ISN 0025            NX = NX - 1
 ISN 0026            RETURN
 ISN 0027           20 IF(VEH(3,MJ).NE.0)  GO TO 100
 ISN 0029            IF(KNS.EQ.3)  GO TO 100
 ISN 0031            CF1 = EXP(B1(MJ))
 ISN 0032            KNS = 2
 ISN 0033            GO TO 100
 ISN 0034           30 IF(VEH(4,MJ).NE.0)  GO TO 100
 ISN 0036            CF1 = EXP(B1(MJ))
```

```
ISN 0037            KNS = 3
ISN 0038          100 CONTINUE
ISN 0039          101 GO TO (200,300,400), KNS
ISN 0040          200 M=0
ISN 0041            N=2
ISN 0042            IF(VEH(3,KX).NE.0) N=3
ISN 0044            IF(VEH(4,KX).NE.0) N=4
ISN 0046            GO TO 500
ISN 0047          300 N=2
ISN 0048            M=1
ISN 0049            IF(VEH(4,KX).NE.0) M=2
ISN 0051            GO TO 500
ISN 0052          400 N=3
ISN 0053            M=1
ISN 0054          500 K1=VEH(1,KX)
ISN 0055            DO 501 I=1,4
ISN 0056            IF(VEH(I,KX).EQ.0) GO TO 502
ISN 0058            K=VEH(I,KX)
ISN 0059            WF(I)=WTFU(K)
ISN 0060            WT(I)=WTIN(K)
ISN 0061            ISPA(I)=ISP(K)
ISN 0062            THUT(I)=THRT(K)
ISN 0063          501 CONTINUE
ISN 0064          502 DO 503 I=1,NMIS
ISN 0065            LZ(I) = 0
ISN 0066            PR=CF1
ISN 0067            VDES=VLR(I)
ISN 0068            CALL PERFI(CLI,N,VREF)
ISN 0069            IF(WPL.GT.WPR(I).AND.IERR.EQ.0) LZ(I) = 1
ISN 0071          503 CONTINUE
ISN 0072            CALL PACK(LZ,VM(1,KX),NMIS,1)
ISN 0073            RETURN
ISN 0074            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000       NAME MOX02MI(R)
IEW0461  PACK
IEW0461  PERFI
IEW0461  EXP
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH
MISMTI	00	466
SAVE1	468	FC4
SVACAV	1430	B48
SAV3	1F78	980
SAVALL	28F8	3A1C
SCRACH	6318	6A60

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
108	SAVE1	SAVE1	10C	SVACAV	SVACAV
110	SAV3	SAV3	114	SAVALL	SAVALL
118	SAVALL	SAVALL	11C	SCRACH	SCRACH
120	PACK	$UNRESOLVED	124	PERFI	$UNRESOLVED
128	EXP	$UNRESOLVED	98	SCRACH	SCRACH
A0	SAV3	SAV3			

ENTRY ADDRESS 00
TOTAL LENGTH C078

```
****MOX02MI   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                    DATE  71.312/17.05.21

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
   ISN 0002            SUBROUTINE NDTR(X,P,D)
                       C
                       C  THIS SUBROUTINE COMPUTES Y = P(X) = PROB THAT THE RANDOM VARIABLE U,
                       C   DISTRIBUTED NORMALLY(0,1) IS LESS THAN OR EQUAL TO X,.  F(X) - THE
                       C ORDINATE OF THE NORMAL DENSITY AT X, IS ALSO COMPUTED.
                       C  DESCRIPTION OF PARAMETERS   X - - INPUT SCALAR FOR WHICH P(X) IS COMPUTED
                       C   P - - OUTPUT PROBABILITY,   D - - OUTPUT DENSITY
                       C  METHOD - - BASED ON APPROX IN C. HASTINGS, APPROXIMATION FOR DIGITAL
                       C   COMPUTERS, PRINCETON UNIV. PRESS, PRINCETON, N.J., 1955.  SEE EQN. 26.2.17,
                       C   HANDBOOK OF MATHEMATICAL FUNCTIONS, ABRAMOWITZ AND STEGUN, DOVER PUBL., INC.
                       C
   ISN 0003              AX = ABS(X)
   ISN 0004              T = 1.0/(1.0 & .2316419*AX)
   ISN 0005              D = 0.3989423*EXP(-X*X/2.0)
   ISN 0006              P = 1.0 - D*T*((((1.330274*T - 1.821256)*T &1.781478)*T
                        *  -0.3565638)*T & .03193815)
   ISN 0007              IF(X) 1,2,2
   ISN 0008             1 P = 1.0 - P
   ISN 0009             2 RETURN
   ISN 0010              END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)              DEFAULT OPTION(S) USED
IEW0000       NAME MOX02NR(R)
IEW0461  EXP


                                        CROSS REFERENCE TABLE

  CONTROL SECTION                       ENTRY

    NAME     ORIGIN  LENGTH               NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

  NDTR           00     1DC


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

       C8            EXP            $UNRESOLVED
 ENTRY ADDRESS       00
 TOTAL LENGTH       1E0

****MOX02NR   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                  DATE  71.312/17.04.53

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
   ISN 0002            SUBROUTINE NDTRI(P,X,C,IE)
                       C
                       C     COMPUTES X = P**(-1)(Y), THE ARGUMENT X SUCH THAT Y = P(X) =
                       C    THE PROB THAT THE RANDOM VARIABLE U, DISTRIBUTED NORMALLY(0,1), IS
                       C     LESS THAN OR EQUAL TO X.   F(X), THE ORDINATE O F THE NORMAL DENSITY,AT X,
                       C     IS ALSO COMPUTED
                       C
                       C   P  -  INPUT PROBABILITY
                       C   X  -  OUTPUT ARGUMENT SUCH  THAT P = Y = THE PROB THAT U, THE RANDOM
                       C         VARIABLE, IS LESS THAN OR EQUAL TO X
                       C   C  -  OUTPUT DENSITY, F(X)
                       C   IER - OUTPUT ERROR CODE
                       C      MAXIMUM ERROR IS 0.00045
                       C
   ISN 0003                    IE = 0
   ISN 0004                    X = .99999E&74
   ISN 0005                    C = X
   ISN 0006                    IF (P) 1,4,2
   ISN 0007                  1 IE = -1
   ISN 0008                    RETURN
   ISN 0009                  2 IF (P-1.0) 7,5,1
   ISN 0010                  4 X = -.999999E&74
   ISN 0011                  5 C = 0.0
   ISN 0012                    RETURN
                       C
   ISN 0013                  7 C = P
   ISN 0014                    IF(C - 0.5) 9,9,8
   ISN 0015                  8 C = 1.0 - C
   ISN 0016                  9 T2 = ALOG(1.0/(C*C))
   ISN 0017                    T = SQRT(T2)
   ISN 0018                    X = T-(2.515517&0.802853*T&0.010328*T2)/(1.0&1.432788*T&
                             1     0.189269*T2 & 0.001308*T*T2)
   ISN 0019                    IF(P-0.5) 10,10,11
   ISN 0020                 10 X = -X
   ISN 0021                 11 C = 0.3989423*EXP(-X*X/2.0)
   ISN 0022                    RETURN
   ISN 0023                    END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
           VARIABLE OPTIONS USED - SIZE=(126976,24576)                DEFAULT OPTION(S) USED
IEW0000      NAME MOX02NI(R)
IEW0461  EXP
IEW0461  SQRT
IEW0461  ALOG

                                                CROSS REFERENCE TABLE

  CONTROL SECTION                          ENTRY
    NAME     ORIGIN  LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME
  NDTRI          00     2A6

  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
       E0           EXP          $UNRESOLVED                  E4           SQRT          $UNRESOLVED
       E8           ALOG         $UNRESOLVED
 ENTRY ADDRESS      00
 TOTAL LENGTH      2A8

****MOX02NI   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                  DATE  71.312/17.12.46

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
  ISN 0002            SUBROUTINE OUTPTI
                  C   ***  PRINT OUT BEST ASSIGNMENT  ***
                  C
  ISN 0003            DOUBLE PRECISION NAME
  ISN 0004            REAL NPERPD,LEVEL
  ISN 0005            LOGICAL EXT,ACCL
  ISN 0006            INTEGER*2  YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,KODE,
                     1  NYRSFX,KODEM,KODESP,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,
                     2   LABI,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,LTR
                  C
  ISN 0007            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                     1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                     2  RPLM(50),MAS(40,3), RXD(12,50)
  ISN 0008            COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                     1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                     2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                     3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
                     4,  R(84),  S(84),CS(90),NPROG(90),KPROG(90),  KODE(90)
  ISN 0009            COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                     1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
  ISN 0010            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                     1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                     2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
                  C
  ISN 0011            DATA ETR/1HE/
  ISN 0012            DATA WTR/1HW/
                  C
  ISN 0013            WRITE (6,4010)
  ISN 0014            DO 805 J=1,NM
  ISN 0015            L=LETT(J)
  ISN 0016            K=LYR(J)
  ISN 0017            M=1899&ILY&K
  ISN 0018            IF(YRLM(J).GT..001)  GO TO 804
  ISN 0020            IF(LETT(J-1).NE.L) WRITE(6,206) NAME(L),VLR(L),WPR(L),RPLM(L),M,
                     1   YRLM(J)
  ISN 0022            IF(LETT(J-1).EQ.L) WRITE(6,2061) M,YRLM(J)
  ISN 0024            GO TO 805
  ISN 0025       804 I = MIN(J)
  ISN 0026            IA=VEH(1,I)
  ISN 0027            IB=VEH(2,I)
  ISN 0028            IC=VEH(3,I)
  ISN 0029            ID=VEH(4,I)

  ISN 0030            X = NMULT(I,L)
  ISN 0031            X = YRLM(J)*X
  ISN 0032            TR = ETR
  ISN 0033            IF(LTR(L).EQ.2)  TR = WTR
  ISN 0035            IF(LETT(J-1).NE.L)  WRITE(6,202)NAME(L),VLR(L),WPR(L),RPLM(L),M,
                     1 X,STG(IA),STG(IB),STG(IC),STG(ID),TR
  ISN 0037            IF(LETT(J-1).EQ.L) WRITE(6,2021) M,X,        STG(IA),STG(IB),
                     1  STG(IC),STG(ID),TR
  ISN 0039       *805 CONTINUE
  ISN 0040            RETURN
  ISN 0041        202 FORMAT (1X,A6,6X,F10.0,4X,F10.0,F10.0,5X,I4,4X,F5.2,9X,5(A4,1X))
  ISN 0042        206 FORMAT (1X,A6,6X,F10.0,4X,F10.0,   F10.0,5X,I4,4X,F5.2,9X,
                     1 32HNO LAUNCH VEHICLE CAN ACCOMPLISH)
  ISN 0043       2021 FORMAT (52X,I4,4X,F5.2,9X,5(A4,1X))
  ISN 0044       2061 FORMAT (52X,I4,4X,F5.2,9X,32HNO LAUNCH VEHICLE CAN ACCOMPLISH)
  ISN 0045       4010 FORMAT(8HOMISSION,4X,14HCHARACTERISTIC,4X,7HPAYLOAD,4X,6HRETURN,
                     1 4X,6HLAUNCH,4X,6HNUMBER,10X,7HOPTIMUM,8X,6HLAUNCH/7H  TITLE,4X,
                     2 16HVELOCITY(FT/SEC),4X,5H(LBS),4X,7HPAYLOAD,5X,4HYEAR,3X,11HOF LA
                     3UNCHES,4X,14HLAUNCH VEHICLE,5X,4HSITE//)
  ISN 0046            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOX02OI(R)
IEW0461  IBCOM=
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH
OUTPTI	00	58C
SAVE1	590	FC4
SAV2	1558	FE0
SAV3	2538	980
SAVALL	2EB8	3A1C

ENTRY

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
270	SAVE1	SAVE1	274	SAV2	SAV2
278	SAV3	SAV3	27C	SAVALL	SAVALL
280	SAVALL	SAVALL	284	IBCOM=	$UNRESOLVED

```
ENTRY ADDRESS       00
TOTAL LENGTH      68DB

****MOX02OI   NOW REPLACED IN DATA SET
```

EXTERNAL SYMBOL DICTIONARY

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
PACK	SD	01	000000	0000E8	
UNPACK	LD		000052		01
ITEM	LD		000096		01

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01FE8

                                    1 *       SUBROUTINE  PACK ( L, M, I, N )
                                    2 *
                                    3 *             THIS ROUTINE PACKS I WORDS IN THE L ARRAY  TO THE
                                    4 *             ARRAY M.  DATA ITEMS L ARE TRUNCATED ON THE LEFT
                                    5 *             AND ONLY THE N LOW ORDER BITS ARE RETAINED.
                                    6 *             PACKED DATA IN M IS LEFT JUSTIFIED WITH 32/N ITEMS
                                    7 *             PER WORD.
                                    8 *
000000                              9 PACK    CSECT
000000                             10         USING  *,15          USE REG 15 FOR BASE
000000 9027 D01C        0001C      11         STM    2,7,28(13)    SAVE REGS
000004 9825 1000        00000      12         LM     2,5,0(1)      LOAD ADDRESSES OF ARGUMENTS
000008 5844 0000        00000      13         L      4,0(4)        I TO REG 4 - NO. OF ITEMS TO BE PACKED
00000C 5875 0000        00000      14         L      7,0(5)        N TO REG 7 - NO. OF BITS/ITEM
000010 4270 F029        00029      15         STC    7,SHIFT+3     MODIFY SHIFT INST WITH NO. OF BITS
000014 1367                        16         LCR    6,7           NO. OF BITS SHIFT FOR DECREMENT
000016 0670                        17         BCTR   7,0           N-1 FOR COMPARAND
000018 1811                        18         SR     1,1           ZERO REG 1
00001A 4150 0020        00020      19 WORD    LA     5,32          LOAD A 32 TO REG 5 FOR COUNT
00001E 5013 0000        00000      20         ST     1,0(3)        ZERO STORAGE AREA
000022 5802 0000        00000      21 LOOP    L      0,0(2)        LOAD DATA TO REG 0
000026 8C00 0000        00000      22 SHIFT   SRDL   0,0           SHIFT DATA TO REG 1
00002A 1800                        23         SR     0,0           TRUNCATE ON LEFT FOR MOD 2**N
00002C 8D00 5000        00000      24         SLDL   0,0(5)        SHIFT BACK TO PROPER POSITION
000030 5603 0000        00000      25         O      0,0(3)        OR PACKED WORD TO REG 0
000034 5003 0000        00000      26         ST     0,0(3)        STORE BACK TO PACKED AREA
000038 4122 0004        00004      27         LA     2,4(2)        INCREMENT DATA ADDRESS
00003C 4640 F046        00046      28         BCT    4,NEXT        COUNT DOWN ON NO. OF ITEMS
000040 9827 D01C        0001C      29         LM     2,7,28(13)    RESTORE REGS
000044 07FE                        30         BR     14            RETURN
000046 8656 F022        00022      31 NEXT    BXH    5,6,LOOP      BRANCH BACK IF SPACE LEFT
00004A 4133 0004        00004      32         LA     3,4(3)        OTHERWISE INCREMENT STORAGE ADDRESS
00004E 47F0 F01A        0001A      33         B      WORD          AND CONTINUE
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01FE8

                                   35 *       SUBROUTINE  UNPACK ( L, M, I, N )
                                   36 *
                                   37 *             THIS ROUTINE UNPACKS I WORDS OF DATA FROM THE M
                                   38 *             ARRAY TO THE L ARRAY.  WORDS IN L ARE ZEROED AND N
                                   39 *             BITS ARE PLACED RIGHT JUSTIFIED FROM THE PACKED
                                   40 *             ARRAY M.
                                   41 *
                                   42         ENTRY  UNPACK
000052                             43         USING  *,15          USE REG 15 FOR BASE
000052 9027 D01C        0001C      44 UNPACK  STM    2,7,28(13)    SAVE REGS
000056 9825 1000        00000      45         LM     2,5,0(1)      LOAD ADDRESSES OF ARGUMENTS
00005A 5844 0000        00000      46         L      4,0(4)        I TO REG 4 - NO. OF ITEMS TO BE PACKED
00005E 5875 0000        00000      47         L      7,0(5)        N TO REG 7 - NO. OF BITS/ITEM
000062 4270 F025        00077      48         STC    7,LEFT+3      MODIFY SHIFT INST WITH NO. OF BITS
000066 1367                        49         LCR    6,7           NO. OF BITS SHIFT FOR DECREMENT
000068 0670                        50         BCTR   7,0           N-1 FOR COMPARAND
00006A 4150 0020        00020      51 DATA    LA     5,32          LOAD A 32 TO REG 5 FOR COUNT
00006E 5813 0000        00000      52         L      1,0(3)        LOAD PACKED DATA TO REG 1
000072 1800                        53 BACK    SR     0,0           ZERO REG 0
000074 8D00 0000        00000      54 LEFT    SLDL   0,0           SHIFT N BITS TO REG 0
000078 5002 0000        00000      55         ST     0,0(2)        STORE IN L
00007C 4122 0004        00004      56         LA     2,4(2)        INCREMENT STORAGE ADDRESS
000080 4640 F038        0008A      57         BCT    4,MORE        COUNT DOWN ON NO. OF ITEMS
000084 9827 D01C        0001C      58         LM     2,7,28(13)    RESTORE REGS
000088 07FE                        59         BR     14            RETURN
00008A 8656 F020        00072      60 MORE    BXH    5,6,BACK      BRANCH BACK IF MORE DATA
00008E 4133 0004        00004      61         LA     3,4(3)        OTHERWISE INCREMENT DATA ADDRESS
000092 47F0 F018        0006A      62         B      DATA          AND CONTINUE
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01FEB
                                    64 *        FUNCTION  ITEM ( M, I, N )
                                    65 *
                                    66 *              THIS ROUTINE RETRIEVES THE I TH ITEM FROM THE PACKED
                                    67 *              ARRAY M.
                                    68 *
                                    69          ENTRY  ITEM
000096                              70          USING  *,15
000096 9025 D01C          0001C     71 ITEM     STM    2,5,28(13)     SAVE REGS
00009A 9824 1000          00000     72          LM     2,4,0(1)       LOAD ADDRESSES OF ARGS TO REGS 2,3,4.
00009E 5833 0000          00000     73          L      3,0(3)         LOAD I TO REG 3
0000A2 0630                         74          BCTR   3,0            SUBTRACT 1 FOR I-1
0000A4 4100 0020          00020     75          LA     0,32           LOAD A 32 TO REG 0
0000A8 8E00 0020          00020     76          SRDA   0,32           SHIFT TO REG 1
0000AC 5D04 0000          00000     77          D      0,0(4)         DIVIDE BY N
0000B0 5010 F04E          000E4     78          ST     1,TEMP         NO. OF ITEMS/WORD
0000B4 1803                         79          LR     0,3            I-1 TO REG 0
0000B6 8E00 0020          00020     80          SRDA   0,32           SHIFT TO REG 1
0000BA 5D00 F04E          000E4     81          D      0,TEMP         DIVIDE I-1 BY NO. ITEMS/WORD
0000BE 1851                         82          LR     5,1            SAVE IN REG 5 TO INDEX ARRAY M
0000C0 8B50 0002          00002     83          SLA    5,2            MULTIPLY BY 4
0000C4 8E00 0020          00020     84          SRDA   0,32           REMAINDER TO REG 1
0000C8 5C04 0000          00000     85          M      0,0(4)         MULTIPLY BY N
0000CC 1831                         86          LR     3,1            LOAD TO REG 3 TO INDEX SHIFT
0000CE 5815 2000          00000     87          L      1,0(5,2)       LOAD DATA FROM M ARRAY
0000D2 8910 3000          00000     88          SLL    1,0(3)         LEFT ADJUST PROPER ITEM
0000D6 5844 0000          00000     89          L      4,0(4)         LOAD N TO REG 4
0000DA 8D00 4000          00000     90          SLDL   0,0(4)         SHIFT N BITS TO REG 0
0000DE 9825 D01C          0001C     91          LM     2,5,28(13)     RESTORE REGS
0000E2 07FE                         92          BR     14             RETURN
0000E4                              93 TEMP     DS     F
                                    94          END
```

CROSS-REFERENCE

```
SYMBOL    LEN  VALUE  DEFN     REFERENCES

BACK      00002 000072 0053     0060
DATA      00004 00006A 0051     0062
ITEM      00004 000096 0071     0069
LEFT      00004 000074 0054     0048
LOOP      00004 000022 0021     0031
MORE      00004 00008A 0060     0057
NEXT      00004 000046 0031     0028
PACK      00001 000000 0009
SHIFT     00004 000026 0022     0015
TEMP      00004 0000E4 0093     0078  0081
UNPACK    00004 000052 0044     0042
WORD      00004 00001A 0019     0033


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
  120 PRINTED LINES
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)
IEW0000       NAME MOX01PK(R)
****MOX01PK    NOW REPLACED IN DATA SET
```

```
FORTRAN IV G LEVEL 1, MOD 4            PDCSTI            DATE = 71312          18/09/23

0001                SUBROUTINE PDCSTI
                C DETERMINE IF MORE THAN ONE PAD NEEDED AT EACH COMPLEX AND IF PREVIOUSLY
                C    UNCONSIDERED COSTS ARE TO BE ADDED TO TOTAL COST FOUND IN ALGORITHM
                C
0002                REAL NPERPD,NPUSED
0003                INTEGER*2 KOUT,LTR,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,
                   1  NVEH,MATCH,JF,JL,NADD,NX,MINOPT,MORE,NPSTG,NPAD,NPFAM,NFS,
                   2  NPINTL,NPINTU,MAPS,MAPF,MAPI,NINTYR,NTGYTR,MAF,MAIC,FINISH,NSTG,
                   3LABF,LABS,LABI,NFML,NFMU,KODS,MAS
                C
0004                COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                   1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                   2  RPLM(50),MAS(40,3), RXD(12,50)
0005                COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                   1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0006                COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                   *              NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                   1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                   2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                   3    PSTGS(30,10,2)
0007                COMMON/SAVALL/LCK,SLD,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                   1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0008                COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
                   1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
0009                COMMON/SCRACH/EXTRA,NADD,NX,MORE(10),ZKP,WKP,NXKP,LZKP(5),DUME(11)
                   *,   A2,LZ(46),W(500),W2(500),
                   1 TDS(500),WR(499),Z(500),COST(2,250),MINOPT(246,9),NODE(5,500),
                   2  NPOS,SIGSQ(9),ETC(9),
                   4 JF(20),JL(20),MATCH(20),NPUSED(20),NVEH(20,6)
                C
0010                IF(MOS.EQ.1.OR.MOS.EQ.3)  GO TO 362
0011                NOT = 0
                C   MORE = NUMBER OF NODE WHICH HAS BEEN CONSIDERED AS OPT. SOLN.
0012              502 DO 355 I = 1,10
0013                IF (MORE(I).EQ.NX) GO TO 360
0014                IF(MORE(I).NE.0) GO TO 355
0015                MORE(I) = NX
0016                NTEM = NPOS + 1
0017                IF(LP.GT.0) WRITE(6,404) NTEM,NX,W(NX),TDS(NX),Z(NX)
0018                GO TO 356
0019              355 CONTINUE
0020              357 WRITE (6,358)

0021                GUESS = 0.0
0022                RETURN
0023              325 WRITE(6,401) NX,PAD(I),J
0024                Z(NX) = 20.0E30
0025                RETURN
0026              360 DO 361 J = I,9
0027                IF(MORE(J).EQ.0)  GO TO 354
0028              361 MORE(J) = MORE(J+1)
0029                MORE(10) = 0
0030              354 IF(NADD.GE.1.OR.(LCK.EQ.1.AND.IFLAG.EQ.0))  GO TO 359
0031                NEXD = 0
0032                IF(NOT.EQ.1)   GO TO 500
0033              362 NOT = 1
0034              356 EXTRA = 0.0
0035                IF(NP.EQ.0)  GO TO 1
0036                DO 320 I = 1,NP
0037                DO 322 K = 1,MYRS
0038                NPUSED(K) = 0.0
0039                DO 322 J = 1,6
0040              322 NVEH(K,J) = 0
                C  COUNT NUMBER OF LAUNCHES REQUIRED PER PAD PER YEAR
0041                DO 321 J = 1,NM
0042                IF(YRLM(J).LT..001)  GO TO 321
0043                KI= MIN(J)
0044                JA = LETT(J)
0045                L = LTR(JA)
0046                IF (NPAD(L,KI).NE.1) GO TO 321
0047                M = LYR(J)
0048                X = NMULT(KI,JA)
0049                X = YRLM(J)*X
0050                NPUSED(M) = NPUSED(M) + X
0051                DO 323 ME = 1,6
0052                IF(NVEH(M,ME).EQ.KI) GO TO 321
0053                IF (NVEH(M,ME).NE.0) GO TO 323
0054                NVEH(M,ME) = KI
0055                GO TO 321
0056              323 CONTINUE
0057              321 CONTINUE
                C   SEE IF SECOND PAD IS REQUIRED AT ANY FACILITY
0058                DO 331 J = 1,20
0059              331 MATCH(J) = 0
0060                DO 324 J = 1,MYRS
0061                IF (NPUSED(J).GT.2.0*NPERPD(1)) GO TO 325
```

```
0062              IF (NPUSED(J).LE.NPERPD(I)) GO TO 324
0063              DO 326 K = 1,10
0064              IF (NPSTG(I,K).EQ.0) GO TO 329
0065              IF (PSTGD(I,K,2) + PSTGS(I,K,2).LT..001) GO TO 326
0066              DO 327 L = 1,6
0067              IF (NVEH(J,L).EQ.0) GO TO 326
0068              LA = NVEH(J,L)
0069              DO 328 LB = 1,4
0070              IF(VEH(LB,LA).EQ.0) GO TO 327
0071              IF (VEH(LB,LA).NE.NPSTG(I,K)) GO TO 328
0072              DO 330 LC = 1,20
0073              IF (MATCH(LC).EQ.0) GO TO 332
0074              IF (MATCH(LC).EQ.K) GO TO 333
0075              GO TO 330
0076          332 MATCH(LC) = K
0077              JF(LC) = J
0078          333 JL(LC) = J
0079              GO TO 326
0080          330 CONTINUE
0081          328 CONTINUE
0082          327 CONTINUE
0083          326 CONTINUE
0084          329 DO 334 K = 1,5
0085              IF (NPFAM(I,K).EQ.0) GO TO 335
0086              IF (PFAMD(I,K,2) + PFAMS(I,K,2).LT..001) GO TO 334
0087              DO 336 L = 1,6
0088              IF (NVEH(J,L).EQ.0) GO TO 334
0089              LA = NVEH(J,L)
0090              DO 337 LB = 1,4
0091              IF (VEH(LB,LA).EQ.0) GO TO 336
0092              LD = VEH(LB,LA)
0093              DO 338 LC = 1,4
0094              IF (NFS(LD,LC).EQ.0) GO TO 337
0095              IF (NFS(LD,LC).NE.NPFAM(I,K)) GO TO 338
0096              K1 = -K
0097              DO 339 LE = 1,20
0098              IF (MATCH(LE).EQ.0) GO TO 340
0099              IF (MATCH(LE).EQ.K1) GO TO 341
0100              GO TO 339
0101          340 MATCH(LE) = K1
0102              JF(LE) = J
0103          341 JL(LE) = J
0104              GO TO 334
0105          339 CONTINUE
0106          338 CONTINUE
0107          337 CONTINUE
0108          336 CONTINUE
0109          334 CONTINUE
0110          335 DO 342 K = 1,5
0111              IF (NPINTL(I,K).EQ.0) GO TO 324
0112              IF (PINTS(I,K,2).LT..001) GO TO 342
0113              DO 343 L = 1,6
0114              IF (NVEH(J,L).EQ.0) GO TO 342
0115              LA = NVEH(J,L)
0116              DO 344 LB = 1,3
0117              IF (VEH(LB+1,LA).EQ.0) GO TO 343
0118              LC = VEH(LB,LA)
0119              DO 345 LD = 1,4
0120              IF (NFS(LC,LD).EQ.0) GO TO 344
0121              IF (NFS(LC,LD).NE.NPINTL(I,K)) GO TO 345
0122              LE = VEH(LB+1,LA)
0123              DO 346 LF = 1,4
0124              IF (NFS(LE,LF).EQ.0) GO TO 345
0125              IF (NFS(LE,LF).EQ.NPINTU(I,K)) GO TO 347
0126          346 CONTINUE
0127              GO TO 345
0128          347 K1 = -100 - K
0129              DO 348 LG = 1,20
0130              IF (MATCH(LG).EQ.0) GO TO 349
0131              IF (MATCH(LG).EQ.K1) GO TO 350
0132              GO TO 348
0133          349 MATCH(LG) = K1
0134              JF(LG) = J
0135          350 JL(LG) = J
0136              GO TO 342
0137          348 CONTINUE
0138          345 CONTINUE
0139          344 CONTINUE
0140          343 CONTINUE
0141          342 CONTINUE
0142          324 CONTINUE
            C     ADD EXTRA PAD COSTS ASSOCIATED WITH THIS SOLUTION
0143              DO 351 J = 1,20
0144              IF (MATCH(J).EQ.0) GO TO 320
0145              IF (MATCH(J).LT.-100) GO TO 352
0146              IF (MATCH(J).LT.0) GO TO 353
```

```
0147              K = MATCH(J)
0148              EXTRA = EXTRA + PSTGD(I,K,2) + PSTGS(I,K,2) * FLOAT(JL(J)-JF(J)+1)
0149              IF(NOT.EQ.0)  GO TO 351
0150              IF(PSTGD(I,K,2) + PSTGS(I,K,2).LT..01)  GO TO 351
0151              NEXD = NEXD + 1
0152              NDUM = NUMD + NEXD
0153              DS(NDUM)          = PSTGD(I,K,2)
0154              SUST(NDUM)          = PSTGS(I,K,2)
0155              MAT(NDUM)         = -300 - I + 2000
0156              MAPS(I,K) = NDUM
0157              NYD(NDUM)         = JF(J)
0158              LYD(NDUM)         = JL(J)
0159              YD(NDUM)          = 1.0
0160              LZ(NDUM)          = JL(J) - JF(J) + 1
0161              NDUM = NDUM + NSPR + NMIS
0162              IS(NDUM)    = JF(J) - 2 + 1900 + ILY
0163              GO TO 351
0164            353 K = -MATCH(J)
0165              EXTRA = EXTRA + PFAMD(I,K,2) + PFAMS(I,K,2)*FLOAT(JL(J)-JF(J)+1)
0166              IF(NOT.EQ.0)  GO TO 351
0167              IF(PFAMD(I,K,2) + PFAMS(I,K,2).LT..01)  GO TO 351
0168              NEXD = NEXD + 1
0169              NDUM = NUMD + NEXD
0170              DS(NDUM)          = PFAMD(I,K,2)
0171              SUST(NDUM)          = PFAMS(I,K,2)
0172              MAT(NDUM)         = -200 - I + 2000
0173              MAPF(I,K) = NDUM
0174              NYD(NDUM)         = JF(J)
0175              LYD(NDUM)         = JL(J)
0176              YD(NDUM)          = 1.0
0177              LZ(NDUM)          = JL(J) - JF(J) + 1
0178              NDUM = NDUM + NSPR + NMIS
0179              IS(NDUM)    = JF(J) - 2 + 1900 + ILY
0180              GO TO 351
0181            352 K = -MATCH(J) - 100
0182              EXTRA = EXTRA + PINTS(I,K,2) * FLOAT(JL(J)-JF(J)+1)
0183              IF(NOT.EQ.0)  GO TO 351
0184              IF(PINTS(I,K,2).LT..001)  GO TO 351
0185              NEXD = NEXD + 1
0186              NDUM = NUMD + NEXD
0187              DS(NDUM)          = 0.0
0188              SUST(NDUM)          = PINTS(I,K,2)
0189              MAT(NDUM)         = -400 -I + 2000
0190              MAPI(I,K) = NDUM
0191              NYD(NDUM)         = JF(J)
0192              LYD(NDUM)         = JL(J)
0193              YD(NDUM)          = 1.0
0194              LZ(NDUM)          = JL(J) - JF(J) + 1
0195              NDUM = NDUM + NSPR + NMIS
0196              IS(NDUM)    = JF(J) - 2 + 1900 + ILY
0197            351 CONTINUE
0198            320 CONTINUE
0199              1 IF(MOS.EQ.1.OR.MOS.EQ.3)  GO TO 501
0200                IF(NOT.EQ.1)   GO TO 500
0201                IF(LP.GT.0)
                   1WRITE(6,403) EXTRA
                C
                C   ADD PREVIOUSLY NEGLECTED SUSTAINING COSTS
0202                IF(LOUT.EQ.0)  GO TO 11
0203            501 DO 10 I = 1,NUMD
0204                IF(KOUT(I).EQ.0.OR.LZ(I).EQ.0)  GO TO 10
0205                LZ(I) = 0
0206                LT = KOUT(I)
0207                DO 8 J = 1,NM
0208                LM = MIN(J)
0209                IF(LM.EQ.0)  GO TO 8
0210                LY = LYR(J)
0211                IF(KI.EQ.2)  LY = (LYR(J) + 1)/KI
0212                L1 = LM
0213                LX = LETT(J)
0214                IF(LTR(LX).EQ.2)  L1 = LM + NV
0215                DO 6 KK = 1,20
0216                IF(NONREC(L1,KK).EQ.0)  GO TO 8
0217                IF(NONREC(L1,KK).NE.I)  GO TO 6
0218                IF(LY.GE.LZ(I))  LZ(I) = LY
0219              6 CONTINUE
0220              8 CONTINUE
0221                IF(MOS.EQ.1.OR.MOS.EQ.3.OR.LZ(I).EQ.0)   GO TO 10
0222                XX = LZ(I)*KI - NYD(I) + 1
0223                EXTRA = EXTRA + XX*SAVS(LT)
0224                WRITE(6,410) I,LZ(I),NYD(I),XX,SAVS(LT),EXTRA
0225            410 FORMAT (3I6,3F10.2)
0226                IF(DS(I).LT.1.0.AND.NSOL.LE.1)
                   1EXTRA = EXTRA + DS(I)
0227             10 CONTINUE
0228                IF(MOS.EQ.1.OR.MOS.EQ.3)  GO TO 500
```

```
FORTRAN IV G LEVEL 1, MOD 4            PDCSTI            DATE = 71312          18/09/23

 0229                CALL PACK(LZ,NODE(1,NX),NUMD,4)
 0230                IF(LP.GT.0)
                    1WRITE (6,405) EXTRA
 0231              11 IF(EXTRA.LT..001)  NOT = 1
 0232                 IF(EXTRA.LT..001)  GO TO 502
 0233                 TDS(NX) = TDS(NX) + EXTRA
 0234                 Z(NX) = Z(NX) + EXTRA
 0235                 IF(LP.GT.0)
                     1WRITE(6,210) Z(NX)
 0236              55 EXTRA = 100.0
 0237                 RETURN
 0238             500 DO 76 NO = 1,NUMD
 0239              76 LZ(NO) = LZ(NO)*KI
 0240                 CALL PACK(LZ,LZOPT(1),  NUMD+NEXD,5)
 0241             359 EXTRA = 0.0
 0242                 RETURN
 0243             210 FORMAT (12H NEW VALUE =, F12.2)
 0244             358 FORMAT(36HOMORE THAN 10 NODES HAVE BEEN TESTED)
 0245             401 FORMAT (26HOPOSSIBLE SOLUTION AT NODE,I4,49H NOT FEASIBLE. MORE TH
                     1AN 2 PADS NEEDED AT COMPLEX,1X,A4,8H IN YEAR, I3)
 0246             403 FORMAT (18HOEXTRA PAD COSTS =, F10.2)
 0247             404 FORMAT (1HO,12(1H*),19H POSSIBLE SOLUTION ,I3,2X,12(1H*)/1H ,I3,
                     1   29X,3(F9.2,5X))
 0248             405 FORMAT (31HOEXTRA PAD & SMALL SUST COSTS =, F10.2)
 0249                 END

TOTAL MEMORY REQUIREMENTS 001BC2 BYTES

 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
           VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000       NAME MOX02PC(R)
IEW0461   IBCOM=
IEW0461   PACK

                                                MODULE MAP

  CONTROL SECTION                          ENTRY
    NAME     ORIGIN   LENGTH                NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

  PDCSTI        00     1BC2
  SAVE1       1BC8      FC4
  SAV3        2B90      980
  SAV4        3510     3188
  SAVALL      6698     3A1C
  TEMP        A0B8     4110
  SCRACH      E1C8     6A60

 ENTRY ADDRESS        00
 TOTAL LENGTH      14C28

****MOX02PC    NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                   DATE  71.312/18.04.59

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
  ISN 0002            SUBROUTINE PERFI(INCL,N,VREF)
  ISN 0003            REAL LENT,ISP,ISPA,INCL
  ISN 0004            INTEGER COUNT
  ISN 0005            COMMON/SCRACH/IP,IV,IG,NPAX(2),NEH(4),NST(41),THRT(41),DIAM(41),
                     1  TSL(41),LENT(41),WTFU(41),WTIN(41),ISP(41),MZ(50),LZ(50),
                     2          WINT(3,60),KX,NX,WGHT(40),WF(4),WT(4),ISPA(4),
                     3  THUT(4),PRT(60),M,VDES,WPL,PR,K1,IERR,DUM(6067)
  ISN 0006            DATA PI,G /3.141593,32.174/
  ISN 0007            K = N
  ISN 0008            TLS = TSL(K1)
  ISN 0009            DIA = DIAM(K1)
  ISN 0010            KODE = 0
  ISN 0011            IERR = 0
  ISN 0012            V = VDES
  ISN 0013            WPL = 0.
  ISN 0014            IF(PR.LE.0.)   GO TO 30
  ISN 0016       10   WPL = PR
  ISN 0017       30   DO 100 COUNT = 1,50
  ISN 0018            W = WPL
  ISN 0019            DO 40 I=1,K
  ISN 0020       40   W = W&WF(I)&WT(I)
  ISN 0021            WLO = W
  ISN 0022            VIDL = 0.
  ISN 0023            DO 50 I=1,K
  ISN 0024            WB = W-WF(I)
  ISN 0025            IF(WB.GT.0.)  GO TO 48
  ISN 0027       45   IERR = 2
  ISN 0028            RETURN
  ISN 0029       48   VIDL = VIDL & G*ISPA(I)*ALOG(W/WB)
  ISN 0030       50   W = WB-WT(I)
  ISN 0031            IF (KODE) 90,55,90
  ISN 0032       55   IF(PR)  70,70,60
  ISN 0033       60   VLOSS = VIDL-VREF
  ISN 0034            GO TO 97
  ISN 0035       70   TB = 0.
  ISN 0036            DO 80 I=1,K
  ISN 0037       80   TB = TB & WF(I)*ISPA(I)/THUT(I)
  ISN 0038            WX = WB * EXP (VDES/ISPA(K)/G)
  ISN 0039            TB = TB - (WX-WB)*ISPA(K)/THUT(K)
  ISN 0040            TOW = TLS/WLO
  ISN 0041            IF (TOW.GT.1.5) TOW = 1.5
  ISN 0043            VLOSS = 6800.*(2.-TOW)  &  2800.*(THUT(1)/TLS-1.)
                     *  &.5E6*PI*DIA**2/WLO&4.1* EXP(TB/125.)  -1530.*COS(INCL/57.296)
  ISN 0044       90   DELV = VIDL-VLOSS-VREF-V
  ISN 0045            WPL = WPL&(WPL&WT(K))*DELV/ISPA(K)/G*(1.&WB/(WB&WF(K)))
  ISN 0046            IF(ABS(DELV).GT.1.)   GO TO 100
  ISN 0048       95   IF(KODE.EQ.0)  GO TO 96
  ISN 0050       110  IF (WPL.LT.0.) IERR = 1
  ISN 0052       120  RETURN
  ISN 0053       96   PR  = WPL
  ISN 0054       97   V = VDES
  ISN 0055            K = N&M
  ISN 0056            KODE = 1
  ISN 0057            WPL = 0.
  ISN 0058       100  CONTINUE
  ISN 0059            IERR = 3
  ISN 0060            RETURN
  ISN 0061            END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
        VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000       NAME MOX02PI(R)
IEW0461  COS
IEW0461  EXP
IEW0461  ALOG

                                                CROSS REFERENCE TABLE

  CONTROL SECTION                          ENTRY

    NAME     ORIGIN  LENGTH                 NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

  PERFI          00     4BC
  SCRACH        4C0    6A60


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      120           SCRACH          SCRACH                   124           COS             $UNRESOLVED
      128           EXP             $UNRESOLVED              12C           ALOG            $UNRESOLVED

 ENTRY ADDRESS         00
 TOTAL LENGTH        6F20

****MOX02PI    NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                DATE  71.312/17.26.58

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
   ISN 0002            SUBROUTINE PRINTI
                  C    ***PRINT OUT DECISION COST CATEGORIES***
                  C
   ISN 0003            REAL NPERPD
   ISN 0004            INTEGER*2 LSA,NYS,KODEF,LST,MST,IST,JST,KST,VEH,NMULT,NONREC,NYD,
                      1  IS,MAT,LYR,LETT,LYD,MIN,LTR,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,
                      2  MAPS,MAPF,MAPI,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,
                      3  MAF,MAIC
   ISN 0005            COMMON/SAVDMP/ NFAM,KFLAG,FAM(30),KODEF(30),FMNR(30),FMSUS(30),
                     1JST(30),YDF(30),LSA(40),SNR(40),NYS(40),DINT(40),SINT(40),KST(40),
                      2  YDI(40),YDS(40),IST(40),FMSLS(30,2),SUSLS(40,2),SINTLS(40,2),
                      3  LST(30,5),YDPF(30,5),MST(30,10),YDPS(30,10)
   ISN 0006            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                     1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                      2  RPLM(50),MAS(40,3), RXD(12,50)
   ISN 0007            COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                     1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
   ISN 0008            COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                      *             NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                     1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                      2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                      3   PSTGS(30,10,2)
   ISN 0009            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                     1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                      2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
                  C
   ISN 0010          2 IF(NUMD.EQ.0)  RETURN
   ISN 0012            WRITE(6,211)
   ISN 0013            DO 925 I = 1,NUMD
   ISN 0014            J=MAT(I)
   ISN 0015            IF(J.GT.1000)  J = J - 2000
   ISN 0017            IF (J.LT.-400) GO TO 9071
   ISN 0019            IF (J.LT.-300) GO TO 9072
   ISN 0021            IF (J.LT.-200) GO TO 9018
   ISN 0023            IF(J.LT.-100) GO TO 345
   ISN 0025            IF(J.LT.0) GO TO 340
   ISN 0027            DO 9073 K = 1,3
   ISN 0028            IF (MAS(J,K).NE.1) GO TO 9073
   ISN 0030            IF(K.EQ.1)  WRITE(6,208)  I,DS(I),SUST(I),STG(J),NYD(I),LYD(I),
                     1  IS(I&NMIS&NSPR),  YD(I)
   ISN 0032            IF(K.EQ.2)  WRITE(6,9074) I,DS(I),SUST(I),STG(J),NYD(I),LYD(I),
                     1  IS(I&NMIS&NSPR),  YD(I)
```

```
ISN 0034            IF(K.EQ.3)  WRITE(6,9075) I,DS(I),SUST(I),STG(J),NYD(I),LYD(I),
                   1  IS(I&NMIS&NSPR),  YD(I)
ISN 0036            GO TO 925
ISN 0037          9073 CONTINUE
ISN 0038           340 JX=-J
ISN 0039            DO 9076 K = 1,3
ISN 0040            IF (MAF(JX,K).NE.I) GO TO 9076
ISN 0042            IF(K.EQ.1)  WRITE(6,209)  I,DS(I),SUST(I),FAM(JX),NYD(I),LYD(I),
                   1   IS(I&NMIS&NSPR), YD(I)
ISN 0044            IF(K.EQ.2)  WRITE(6,9077) I,DS(I),SUST(I),FAM(JX),NYD(I),LYD(I),
                   1   IS(I&NMIS&NSPR), YD(I)
ISN 0046            IF(K.EQ.3)  WRITE(6,9078) I,DS(I),SUST(I),FAM(JX),NYD(I),LYD(I),
                   1   IS(I&NMIS&NSPR), YD(I)
ISN 0048            GO TO 925
ISN 0049          9076 CONTINUE
ISN 0050           345 JX=-J-100
ISN 0051            JY=NFML(JX)
ISN 0052            JZ=NFMU(JX)
ISN 0053            DO 9079 K = 1,3
ISN 0054            IF (MAIC(JX,K).NE.I) GO TO 9079
ISN 0056            IF(K.EQ.1)  WRITE(6,210)   I,DS(I),SUST(I),FAM(JY),FAM(JZ),NYD(I),
                   1   LYD(I), IS(I&NMIS&NSPR),  YD(I)
ISN 0058            IF(K.EQ.2)  WRITE(6,9080)  I,DS(I),SUST(I),FAM(JY),FAM(JZ),NYD(I),
                   1   LYD(I), IS(I&NMIS&NSPR),  YD(I)
ISN 0060            IF(K.EQ.3)  WRITE(6,9081)  I,DS(I),SUST(I),FAM(JY),FAM(JZ),NYD(I),
                   1   LYD(I), IS(I&NMIS&NSPR),  YD(I)
ISN 0062            GO TO 925
ISN 0063          9079 CONTINUE
ISN 0064          9018 JX = -J - 200
ISN 0065            DO 9082 K = 1,5
ISN 0066            IF (MAPF(JX,K).NE.I) GO TO 9082
ISN 0068            KX = NPFAM(JX,K)
ISN 0069            WRITE(6,9083) I,DS(I),SUST(I),FAM(KX),PAD(JX),NYD(I),LYD(I),
                   1    IS(I&NMIS&NSPR),   YD(I)
ISN 0070            GO TO 925
ISN 0071          9082 CONTINUE
ISN 0072          9072 JX = -J - 300
ISN 0073            DO 9084 K = 1,10
ISN 0074            IF (MAPS(JX,K).NE.I) GO TO 9084
ISN 0076            KX = NPSTG(JX,K)
ISN 0077            WRITE(6,9085) I,DS(I),SUST(I),STG(KX),PAD(JX),NYD(I),LYD(I),
                   1    IS(I&NMIS&NSPR),   YD(I)

ISN 0078            GO TO 925
ISN 0079          9084 CONTINUE
ISN 0080          9071 JX = -J - 400
ISN 0081            DO 9086 K = 1,5
ISN 0082            IF (MAPI(JX,K).NE.I) GO TO 9086
ISN 0084            KX = NPINTL(JX,K)
ISN 0085            KY = NPINTU(JX,K)
ISN 0086            WRITE(6,9087)I,DS(I),SUST(I),FAM(KX),FAM(KY),PAD(JX),NYD(I),LYD(I)
                   1 ,  IS(I&NMIS&NSPR),   YD(I)
ISN 0087            GO TO 925
ISN 0088          9086 CONTINUE
ISN 0089           925 CONTINUE
ISN 0090            RETURN
ISN 0091           208 FORMAT (I4,6X,2F12.2,5X,A4,1X,14HSTAGE HARDWARE,29X,I3,9X,I3,8X,
                   1   I5,7X,F5.0)
ISN 0092           209 FORMAT (I4,6X,2F12.2,5X,A4,1X,15HSHARED HARDWARE,28X,I3,9X,I3,8X,
                   1    I5,7X,F5.0)
ISN 0093           210 FORMAT(I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,
                   1  9H HARDWARE,11X,I3,9X,I3,8X,I5,7X,F5.0)
ISN 0094           211 FORMAT(25H1QUANTITIES BRANCHED UPON/1HO,6HNUMBER,5X,11HDEVELOPMENT
                   *, 2X, 10HSUSTAINING,50X,10HYEAR AVAIL,2X,9HLAST YEAR,2X,9HDEV STAR
                   *T, 2X, 12HDEV DURATION//)
ISN 0095           9074 FORMAT (I4,6X,2F12.2,5X,A4,1X,9HSTAGE ETR,34X,I3,9X,I3,8X,I5,7X,
                   *    F5.0)
ISN 0096           9075 FORMAT (I4,6X,2F12.2,5X,A4,1X,9HSTAGE WTR,34X,I3,9X,I3,8X,I5,7X,
                   *    F5.0)
ISN 0097           9077 FORMAT (I4,6X,2F12.2,5X,A4,1X,10HSHARED ETR,33X,I3,9X,I3,8X,I5,
                   *       7X, F5.0)
ISN 0098           9078 FORMAT (I4,6X,2F12.2,5X,A4,1X,10HSHARED WTR,33X,I3,9X,I3,8X,I5,7X,
                   *    F5.0)
ISN 0099           9080 FORMAT (I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,4H ETR,
                   1  16X,I3,9X,I3,8X,I5,7X,F5.0)
ISN 0100           9081 FORMAT (I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,4H WTR,
                   1  16X,I3,9X,I3,8X,I5,7X,F5.0)
ISN 0101           9083 FORMAT (I4,6X,2F12.2,5X,A4,1X,14HSHARED AT PAD ,A4,25X,I3,9X,I3,
                   *    8X,I5,7X,F5.0)
ISN 0102           9085 FORMAT (I4,6X,2F12.2,5X,A4,1X,13HSTAGE AT PAD ,A4,26X,I3,9X,I3,8X,
                   *    I5,7X,F5.0)
ISN 0103           9087 FORMAT (I4,6X,2F12.2,5X,15HINTEGRATION OF ,A4,5H AND ,A4,1X,
                   1  7HAT PAD ,A4,8X,I3,9X,I3,8X,I5,7X,F5.0)
ISN 0104            END

****** END OF COMPILATION ******
```

```
F8H-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                DEFAULT OPTION(S) USED
IEW0000      NAME MUX02PN(R)
IEW0461   IBCOM=

                                                CROSS REFERENCE TABLE

 CONTROL SECTION                          ENTRY

   NAME     ORIGIN  LENGTH                  NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

 PRINTI         00    108A
 SAVDMP       1090    148C
 SAVE1        2550     FC4
 SAV3         3518     980
 SAV4         3E98    3188
 SAVALL       7020    3A1C


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      410           SAVDMP          SAVDMP                   414           SAVE1           SAVE1
      418           SAV3            SAV3                     41C           SAV4            SAV4
      420           SAV4            SAV4                     424           SAV4            SAV4
      428           SAVALL          SAVALL                   42C           SAVALL          SAVALL
      430           IBCOM=          $UNRESOLVED
ENTRY ADDRESS       00
TOTAL LENGTH      AA40

****MUX02PN   NOW REPLACED IN DATA SET


(17)          OS/360  FORTRAN H                                DATE  71.312/18.10.52

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
ISN 0002            SUBROUTINE REUSE
                C ESTIMATE NUMBER OF INITIAL UNITS TO PURCHASE
                C
ISN 0003            INTEGER*2  NU,NBY,MODE,NOB,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,
                   1  LYD,MIN
ISN 0004            COMMON/SAVSAR/COR,POJ(3),SRJ(3,3),NU(40),NBY(40),NOB(40),RINT(40),
                   1   PLCINT(40),XLT(40), PLCT(40),UPP(40),TAT(40),TAMT(50),SR(40,3),
                   2   MODE(40,3),PLC(40,3)
ISN 0005          .  COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                  1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                  2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
ISN 0006            COMMON/SCRACH/  II,NUS(40),MSAVE(40),ISAVE(40),KLUE(40),
                   1STGYHW(40,20),RINTMX(40,20),STGMAX(40,20,2),STGYTR(40,20,2),
                   2  RINTYR(40,20),DUM(1047)
                C
ISN 0007            I = II
ISN 0008            IF(KLUE(I).GT.0)  GO TO 100
ISN 0010            NU(I) = -  MAXO (2,-NUS(I) -1)
ISN 0011            RETURN
ISN 0012          100 TL = 0.0
ISN 0013            YY = TAT(I)
ISN 0014            NU(I) = -2
ISN 0015          .  DO 200 J = 1,MYRS
ISN 0016            IF(STGYHW(I,J).LT..001)  GO TO 200
ISN 0018            TAM = 365./STGYHW(I,J)
                C   TAM = MAX ALLOWABLE AVERAGE TA TIME IN DAYS FOR YEAR J
ISN 0019            XX = YY
ISN 0020            TL = TL & STGYHW(I,J)
ISN 0021            YY = TAT(I)*(TL)**PLCT(I)
ISN 0022            YY = 2.0 *YY - TAT(I)
ISN 0023            TAA = .5*(XX & YY)
ISN 0024            IF(NOB(I).EQ.1)  GO TO 120
                C  CALCULATE AVERAGE MISSION TA TIME FOR ORBITER ONLY
ISN 0026            COUNT = 0.0
ISN 0027            TOT = 0.0
ISN 0028            DO 110 K = 1,NM
ISN 0029            IF(LYR(K).NE.J)  GO TO 110
                C  CHECK IF STAGE I IS TOP STAGE OF VEHICLE MK
ISN 0031            MK = MIN(K)
ISN 0032            DO 105 I1 = 1,4
ISN 0033            I11 = 5-I1
ISN 0034            IF(VEH(I11 ,MK).EQ.0)  GO TO 105
```

```
ISN 0036            IF(VEH(I11 ,MK).EQ.I)  GO TO 106
ISN 0038            GO TO 110
ISN 0039          105 CONTINUE
ISN 0040          106 NETT = LETT(K)
ISN 0041            XN = NMULT(MK,NETT)
ISN 0042            COUNT = COUNT & YRLM(K)*XN
ISN 0043            TOT = TOT & TAMT(NETT)*XN*YRLM(K)
ISN 0044          110 CONTINUE
ISN 0045            TAA = TAA  & .5 & TOT/COUNT
ISN 0046          120 IF(TAM.GE.TAA)  GO TO 200
ISN 0048            NRQY = TAA/TAM & .9999
ISN 0049            NX = NU(I)
ISN 0050            NU(I) = MINO(-NRQY,NX)
ISN 0051          200 CONTINUE
                C     COMPARE NUMBER REQUIRED BY LIFETIME TO NUMBER REQUIRED BY TAT
ISN 0052            X = - NU(I)
ISN 0053            IF((X*XLT(I)).LT.TL)  NU(I) = - INT(TL/XLT(I) & .9999)
ISN 0055            RETURN
ISN 0056            END

****** END OF COMPILATION ******
```

```
 F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
         VARIABLE OPTIONS USED - SIZE=(126976,24576)                DEFAULT OPTION(S) USED
IEW0000      NAME MOX02RS(R)
IEW0461  FRXPR=


                                                CROSS REFERENCE TABLE

  CONTROL SECTION                          ENTRY

    NAME     ORIGIN  LENGTH                  NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

  REUSE          00     4D8
  SAVSAR        4D8     A5C
  SAVALL        F38    3A1C
  SCRACH       4958    6A60


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      128           SAVSAR         SAVSAR                    12C           SAVALL         SAVALL
      130           SAVALL         SAVALL                    134           SCRACH         SCRACH
      138           SCRACH         SCRACH                    13C           SCRACH         SCRACH
      140           FRXPR=         $UNRESOLVED
 ENTRY ADDRESS       00
 TOTAL LENGTH      B3B8

****MOX02RS   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                DATE  71.312/18.06.21

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
   ISN 0002             SUBROUTINE REVLUS
                      CTHIS SUBROUTINE RECALCULATES THE APPROPRIATE VALUES FOR RECURRING COSTS
   ISN 0003             INTEGER H,PROG
   ISN 0004             INTEGER*2  LTR,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,
                       6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
   ISN 0005          .  LOGICAL SKIP
   ISN 0006             REAL NPERPD
   ISN 0007             COMMON/SAVRT/RVAR(20,50)
   ISN 0008             COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                       1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
   ISN 0009             COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZUPT(8),NYD(46),MAT(46
                       1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                       2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
   ISN 0010             COMMON/VARNCE/KSTAT,VARI(40),VARF(50),VARM(56),FMVAR(2,30),
                       1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)
   ISN 0011             COMMON/SCRACH/M,N,NCS,PROG,IODD,IERR,SKIP,MYFLAG,JS,NSCALE(5),
                       1  NSL(10),TOTAL(20),W(20),D(20),XOUT(20),VOUT(20),RRR(20),YEAR(20)
                       2,  Y(20),KVEHI(50),LABEL(50),LVARY(70),LVD(70),IVEH(70),LVS(70),
                       3  LVSF(80),VNAM(80),NOP(86),RF(86),CF(86),SF(86),FLAGR(86),
                       4  FLAGS(86),NSSF(86),NSRF(86),NSXF(86),NDSF(86),SUSTF(86),NLVP(86)
                       5;  NSTRRC(86),NYRSRC(86),LNDF(86),NSTRST(86),LNDATE(86),NPRO(90),
                       6  KPRO(90),CSX(90),LZ(46),RCOST( 60), KVEH( 60),IMAGE(830),
                       7  XSCH(10,70),PLSCH(10,70),XLVSUM(20,50),RECUR(20,50),DUM(361),
                       8  RVX(20),RPLX(20)
                     C
   ISN 0012             DO 33 II = 1,20
   ISN 0013             RVX(II) = 0.0
   ISN 0014          33 RPLX(II) = 0.0
   ISN 0015             L = PROG
   ISN 0016             NSTRRC(L) = 100
   ISN 0017             NYRSRC(L) = 0
   ISN 0018             LNDATE(L) = 100
   ISN 0019             IF (NLVP(L).EQ.0) GO TO 21
   ISN 0021             DO 34 LC = 1,20
   ISN 0022          34 RECUR(LC,L) = 0.0
   ISN 0023             IJ = NLVP(L)
   ISN 0024             H = LVARY(L)
   ISN 0025             IB = LVS(H)
   ISN 0026             IF (IB.LT.4) IB = 4
   ISN 0028             DO 38 K=1,IJ
   ISN 0029             IF (LVD(H).EQ.0) GO TO 38
   ISN 0031             IA = LVS(H)-3
   ISN 0032             IF (IA.LT.1) IA=1
   ISN 0034             IK = LVD(H)
   ISN 0035             ILV = IVEH(H)
   ISN 0036             DO 37 J=1,IK
                     C  RCST = VEH. RECURRING COST/YR. BY MISSION
   ISN 0037             RCST = XSCH(J,H)*RCOST(ILV)
                     C RCPL = PAYLOAD RECURRING COST/YR.
   ISN 0038             RCPL = PLSCH(J,H)*PLR(L)
   ISN 0039             DO 36 I=1,4
   ISN 0040             II = LVS(H)-IB&I&J-1
   ISN 0041             IF (II.LT.1) II=1
   ISN 0043             IF(KSTAT.EQ.0.OR.SKIP)  GO TO 36
   ISN 0045             RVX(II) = RVX(II) & RCST
   ISN 0046             RPLX(II) = RPLX(II) & RCPL
                     C  DISTRIBUTE RECURRING COST BY YEAR
   ISN 0047          36 RECUR(II,L) = RECUR(II,L)&ALPI(I,ILV)*RCST & RDIST(L,I)*RCPL
   ISN 0048          37 CONTINUE
   ISN 0049             IF(KSTAT.EQ.0.OR.SKIP)  GO TO 42
   ISN 0051             VTOT = 0.0
   ISN 0052             DO 39 JX = 1,4
   ISN 0053             KX = VEH(JX,ILV)
   ISN 0054             IF(KX.EQ.0)  GO TO 40
   ISN 0056          39 VTOT = VTOT & SVAR(1,KX)
   ISN 0057             JX = JX & 1
   ISN 0058          40 XJX = JX - 1
   ISN 0059             DO 41 IX = 1,20
   ISN 0060             RVAR(IX,L) = (VTOT*RVX(IX))/XJX       & PLVAR(1,L)*RPLX(IX)
   ISN 0061             IF(RVAR(IX,L).GT..001)RVAR(IX,L) = RVAR(IX,L)/(RVX(IX) & RPLX(IX))
   ISN 0063          41 CONTINUE
   ISN 0064          42 NYRSRC(L) = MAXO (NYRSRC(L),II)
   ISN 0065             NSTRRC(L) = MINO (NSTRRC(L),IA)
   ISN 0066             LSUB = LNDATE(L)
   ISN 0067             LVSUB = LVS(H)
   ISN 0068             LNDATE(L) = MINO(LSUB,LVSUB)
   ISN 0069          38 H = H & 1
                     C  NYRSRC & NSTRRC = 0 FOR DEVELOPMENT PROGRAMS
   ISN 0070          21 IF (NSTRRC(L).EQ.100) NSTRRC(L) = 0
   ISN 0072          99 RETURN
   ISN 0073             END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000      NAME MOXO2RV(R)


                                                CROSS REFERENCE TABLE

  CONTROL SECTION                          ENTRY

    NAME     ORIGIN  LENGTH                  NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME

  REVLUS         00     672
  SAVRT         678     FA0
  SAV3         1618     980
  SAVALL       1F98    3A1C
  VARNCE       5988     ADC
  SCRACH       6498    6A60


  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION          LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

      128           SAVRT          SAVRT                     12C           SAV3           SAV3
      130           SAVALL         SAVALL                    134           SAVALL         SAVALL
      138           VARNCE         VARNCE                    13C           SCRACH         SCRACH
      140           SCRACH         SCRACH                    144           SCRACH         SCRACH
      148           SCRACH         SCRACH                    14C           SCRACH         SCRACH
      150           SCRACH         SCRACH
ENTRY ADDRESS       00
TOTAL LENGTH      CEF8

****MOXO2RV   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                              DATE  71.312/17.08.46

         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
 ISN 0002            SUBROUTINE SHIFTS
                     C  THIS SUBROUTINE SHIFTS THE DEVELOPMENT START DATES AND DURATION IN ORDER
                     C    TO ACHIEVE A SMOOTHER LEVEL OF SPENDING
                     C
 ISN 0003            DOUBLE PRECISION NAME
 ISN 0004            LOGICAL SKIP,ACCL,EXT
 ISN 0005            REAL LEVEL
 ISN 0006            INTEGER H,PROG
 ISN 0007            INTEGER*2     YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,
                    1  KODE,NYRSFX,KODEM,KODESP,
                    6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
 ISN 0008            COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                    1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                    2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                    3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
                    4,  R(84), S(84),CS(90),NPROG(90),KPROG(90), KODE(90)
 ISN 0009            COMMON/SCRACH/M,N,NCS,PROG,IODD,IERR,SKIP,MYFLAG,JS,NSCALE(5),
                    1  NSL(10),TOTAL(20),W(20),D(20),XOUT(20),VOUT(20),RRR(20),YEAR(20)
                    2,  Y(20),KVEHI(50),LABEL(50),LVARY(70),LVD(70),IVEH(70),LVS(70),
                    3  LVSF(80),VNAM(80),NOP(86),RF(86),CF(86),SF(86),FLAGR(86),
                    4  FLAGS(86),NSSF(86),NSRF(86),NSXF(86),NDSF(86),SUSTF(86),NLVP(86)
                    5,  NSTRRC(86),NYRSRC(86),LNDF(86),NSTRST(86),LNDATE(86),NPRO(90),
                    6  KPRO(90),CSX(90),LZ(46),RCOST( 60), KVEH( 60),IMAGE(830),
                    7  XSCH(10,70),PLSCH(10,70),XLVSUM(20,50),RECUR(20,50),DUM(401)
                     C
 ISN 0010          110 IODD = IODD & 1
 ISN 0011              GO TO (140,150,160,168,170,178,180),  IODD
 ISN 0012          140 STR = S(PROG)
 ISN 0013              S(PROG) = STR & 1.0
 ISN 0014              IF(S(PROG).GT.TREF & 21.)  NOP(PROG) = 1
 ISN 0016          145 CALL CONSTR
 ISN 0017              IF (IERR.NE.0) GO TO 110
 ISN 0019           14 MYFLAG = 1
 ISN 0020              RETURN
 ISN 0021          150 S(PROG) = STR - 1.0
 ISN 0022              IF (S(PROG).LT.TREF) GO TO 110
 ISN 0024              GO TO 145
 ISN 0025          160 S(PROG) =  STR
 ISN 0026              IF(R(PROG).LE..0001.OR.CF(PROG).LE..0001)   GO TO 190
 ISN 0028              CKR = R(PROG)
 ISN 0029              CKC = C(PROG)
 ISN 0030              CKS = SUS (PROG)
```

```
ISN 0031            NDS = NYRSST(PROG)
ISN 0032            NSS = NSTRST(PROG)
ISN 0033            NSR = NSTRRC(PROG)
ISN 0034            NYRC = NYRSRC(PROG)
ISN 0035            NSX = NSTRFX(PROG)
ISN 0036            NSO = LNDATE(PROG)
ISN 0037            IF(.NOT.EXT)  GO TO 110
ISN 0039            R(PROG) = CKR & 1.0
ISN 0040            NSTRST(PROG) = INT(2.0*R(PROG)/3.0 & .999)
ISN 0041            NSTRRC(PROG) = NSR & 1
ISN 0042            NSTRFX(PROG) = NSX & 1
ISN 0043            LNDATE(PROG) = NSO & 1
ISN 0044            IF (NLVP(PROG).EQ.0) GO TO 165
ISN 0046            IJ = NLVP(PROG)
ISN 0047            H = LVARY(PROG)
ISN 0048            DO 162 I=1,IJ
ISN 0049            NSL(I) = LVS(H)
ISN 0050            LVS(H) = LVS(H) & 1
ISN 0051         162 H = H & 1
ISN 0052            DO 34 LC = 1,20
ISN 0053          34 RRR(LC) = RECUR(LC,PROG)
ISN 0054         164 CALL REVLUS
ISN 0055         165 CALL CONSTR
ISN 0056            IF (IERR.NE.0) GO TO 110
ISN 0058            IF(RF(PROG) - R(PROG)) 9010,9020,9030
                 C  DEVELOPMENT DURATION IS STRETCHED OUT
ISN 0059         9010 C(PROG) = (.8 & .2*R(PROG)/RF(PROG)) * CF(PROG)
ISN 0060            GO TO 9050
ISN 0061         9020 C(PROG) = CF(PROG)
ISN 0062            GO TO 9050
                 C  DEVELOPMENT DURATION IS ACCELERATED - CRASH PROGRAM
ISN 0063         9030 X = AINT (.5*RF(PROG) & .99)
ISN 0064            IF(R(PROG).LT.X)  R(PROG) = X
ISN 0066            C(PROG) = CF(PROG) * EXP ((1. -R(PROG)/RF(PROG))/ (R(PROG)/
                   1  RF(PROG) - .4))
ISN 0067         9050 IF (NYRSST(PROG).EQ.0) GO TO 14
ISN 0069            NYRSST(PROG) = NDSF(PROG) - LNDF(PROG) & LNDATE(PROG) -
                   1   NSTRST(PROG) & NSSF(PROG)
                 C  THE FOLLOWING DEFN. OF NYRSST IS THE ORIGINAL
                 C     NYRSST(PROG) = R(PROG)/RF(PROG)*FLOAT(NDSF(PROG))&.001
ISN 0070            X = NDSF(PROG)
ISN 0071            X1 = NYRSST(PROG)
```

```
ISN 0072            SUS (PROG) = C(PROG)/CF(PROG)*SUSTF(PROG)*X/X1
ISN 0073            GO TO 14
ISN 0074         168 IF(.NOT.EXT)  GO TO 110
ISN 0076            S(PROG) = STR - 1.0
ISN 0077            IF(S(PROG).LT.TREF)  GO TO 110
ISN 0079            IF(NLVP(PROG).EQ.0)  GO TO 165
ISN 0081            GO TO 164
ISN 0082         170 S(PROG) = STR
ISN 0083            IF (CKR.EQ.RF(PROG).AND..NOT.ACCL)  GO TO 180
ISN 0085            R(PROG) = CKR - 1.0
ISN 0086            NSTRST(PROG) = INT(2.0*R(PROG)/3.0 & .999)
ISN 0087            NSTRRC(PROG) = NSR - 1
ISN 0088            NSTRFX(PROG) = NSX - 1
ISN 0089            LNDATE(PROG) = NSO - 1
ISN 0090            IF (NLVP(PROG).EQ.0) GO TO 165
ISN 0092            IJ = NLVP(PROG)
ISN 0093            H = LVARY(PROG)
ISN 0094            DO 172 I=1,IJ
ISN 0095            LVS(H) = NSL(I) - 1
ISN 0096         172 H = H & 1
ISN 0097         175 GO TO 164
ISN 0098         178 S(PROG) = STR & 1.0
ISN 0099            IF(S(PROG).GT.TREF & 21.)  NOP(PROG) = 1
ISN 0101            IF(NLVP(PROG).EQ.0)  GO TO 165
ISN 0103            GO TO 164
ISN 0104         180 S(PROG) = STR
ISN 0105            R(PROG) = CKR
ISN 0106            C(PROG) = CKC
ISN 0107            SUS (PROG) = CKS
ISN 0108            NYRSST(PROG) = NDS
ISN 0109            NSTRST(PROG) = NSS
ISN 0110            NSTRRC(PROG) = NSR
ISN 0111            NYRSRC(PROG) = NYRC
ISN 0112            NSTRFX(PROG) = NSX
ISN 0113            LNDATE(PROG) = NSO
ISN 0114            IF (NLVP(PROG).EQ.0) GO TO 190
ISN 0116            IJ = NLVP(PROG)
ISN 0117            H = LVARY(PROG)
ISN 0118            DO 182 I=1,IJ
ISN 0119            LVS(H) = NSL(I)
ISN 0120         182 H = H & 1
ISN 0121            DO 36 LC = 1,20
```

```
ISN 0122        36 RECUR(LC,PROG) = RRR(LC)
ISN 0123       190 MYFLAG = 0
ISN 0124           RETURN
ISN 0125           END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOX02SH(R)
IEW0461  EXP
IEW0461  CONSTR
IEW0461  REVLUS

                                CROSS REFERENCE TABLE

  CONTROL SECTION                    ENTRY
    NAME     ORIGIN  LENGTH          NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME
  SHIFTS       00      7C2
  SAV2        7C8      FE0
  SCRACH     17A8     6A60

  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION        LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
     138          SAV2          SAV2                       13C          SCRACH          SCRACH
     140          SCRACH        SCRACH                     144          SCRACH          SCRACH
     148          SCRACH        SCRACH                     14C          SCRACH          SCRACH
     150          EXP           $UNRESOLVED                154          CONSTR          $UNRESOLVED
     158          REVLUS        $UNRESOLVED
  ENTRY ADDRESS      00
  TOTAL LENGTH     8208

****MOX02SH   NOW REPLACED IN DATA SET
```

```
0001                SUBROUTINE SMOOTH
                C   BUDGET SMOOTHING PROGRAM - R.E. SLYE
                C    MODIFIED BY C.J. GOLDEN
0002                DOUBLE PRECISION NAME
0003                LOGICAL SKIP,OUT,ACCL,EXT
0004                REAL LEVEL,NPERPD
0005                INTEGER PROG,H
0006                INTEGER*2  LTR,YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,
                   1  KODE,NYRSFX,KODEM,KODESP,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,
                   2  LABF,LABI,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,
                   6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
0007                COMMON/SAVER/  RFIXD(12,84)
0008                COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                   1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                   2  PPLM(50),MAS(40,3), RXD(12,50)
0009                COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                   1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                   2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                   3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
                   4,  R(84), S(84),CS(90),NPROG(90),KPROG(90), KODE(90)
0010                COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                   1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0011                COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,L70PT(8),NYD(46),MAT(46
                   1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                   2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0012                COMMON/VARNCE/KSTAT,VARI(40),VARF(50),VARM(56),FMVAR(2,30),
                   1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)
0013                COMMON/SCRACH/M,N,NCS,PROG,IODD,IERR,SKIP,MYFLAG,JS,NSCALE(5),
                   1  NSL(10),TOTAL(20),W(20),D(20),XOUT(20),YOUT(20),RRR(20),YEAR(20)
                   2,  Y(20),KVEHI(50),LABEL(50),LVARY(70),LVD(70),IVEH(70),LVS(70),
                   3  LVSF(80),VNAM(80),NOP(86),RF(86),CF(86),SF(86),FLAGR(86),
                   4  FLAGS(86),NSSF(86),NSRF(86),NSXF(86),NDSF(86),SUSTF(86),NLVP(86)
                   5,  NSTRRC(86),NYRSRC(86),LNDF(86),NSTRST(86),LNDATE(86),NPRO(90),
                   6  KPRO(90),CSX(90),LZ(46),RCOST(60),KVEH(60),IMAGE(830),
                   7  XSCH(10,70),PLSCH(10,70),XLVSUM(20,50),RECUR(20,50),KODX(90),
                   8  NPROD(90),KPROD(90),KODD(90),XMODE(20),UB(20),DUM(1)
0014                DIMENSION PRGLV(4)
0015                EQUIVALENCE (LS,LEVEL(1)),(LF,LEVEL(2))
0016                DATA ASTR /1H*/
0017                DATA ZERO /1HO/
0018                DATA FLET /1HF/
0019                DATA MLET /1HM/
0020                DATA ULET /1HU/
0021                DATA PRGLV /4HPROG, 4HRAM , 4HLEVE, 4HL    /
0022                DATA BLANK /1H /
                C
0023                IODD = 0
0024                NSCALE(1) = 1
0025                NSCALE(2) = 0
0026                NSCALE(3) = 0
0027                NSCALE(4) = 0
0028                NSCALE(5) = 0
0029                IF(FINISH.GT.1)  GO TO 18
0030                PMAX = 5000.
0031                PMIN = 1500.
                C   ACCL = TRUE  IMPLIES USE ACCELERATION OPTION
0032                ACCL = .TRUE.
                C    EXT = TRUE   IMPLIES USE EXTENSION OPTION
0033                EXT = .TRUE.
0034                DO 5 I=1,10
0035               5 TITLE (I) = BLANK
0036                DO 6  I = 1,20
0037                CNTRVL(I) = BLANK
0038               6 FIXED(I) = 0.0
0039                WRITE(6,399)
0040               16 CALL INPUT (6HTITLE , TITLE, 6HLEVEL ,LEVEL, 6HISTRT ,ISTRT,
                     X  6HIFIN  ,IFIN, 6HMAXITR,MAXITR,6HNCSTR ,NCSTR,6HNPROG ,NPROD,
                     X   6HKPROG ,KPROD,6HKODE  ,KODD,6HCS    ,CS,6HFIXED ,FIXED,
                     X   6HPMAX  ,PMAX, 6HPMIN  ,PMIN, 6HACCL  , ACCL, 6HEXT   ,EXT)
0041                DO 550 I = 1,NCSTR
0042                NPROG(I) = NPROD(I)
0043                KPROG(I) = KPROD(I)
0044              550 KODE(I) = KODD(I)
0045                DO 79 I = 1,20
0046               79 FIXED(I) = FIXED(I)*(1. + GRO)**(I-1)
0047                IF(NCSTR.EQ.0)  GO TO 18
0048                DO 8 I = 1,NCSTR
0049                DO 2 I1 = 1,NMIS
0050                IF(NPROG(I).EQ.KODEM(I1))  GO TO 3
0051               2 CONTINUE
0052                GO TO 36
0053               3  NPROG(I) = I1
0054              36 IF(KPROG(I).EQ.0)  GO TO 8
0055                DO 1 I1 = 1,NMIS
0056                IF(KPROG(I).EQ.KODEM(I1))  GO TO 4
0057               1 CONTINUE
```

```
0058              GO TO 8
0059            4 KPROG(I) = I1
0060            8 CONTINUE
0061              IF(NSPR.EQ.0)  GO TO 18
0062              DO 510 I = 1,NCSTR
0063              DO 502 I1 = 1,NSPR
0064              IF(NPROG(I).EQ.KODESP(I1))  GO TO 503
0065          502 CONTINUE
0066              GO TO 37
0067          503 NPROG(I) = I1 + NMIS
0068           37 IF(KPROG(I).EQ.0)   GO TO 510
0069              DO 501 I1 = 1,NSPR
0070              IF(KPROG(I).EQ.KODESP(I1))  GO TO 504
0071          501 CONTINUE
0072              GO TO 510
0073          504 KPROG(I) = I1 + NMIS
0074          510 CONTINUE
0075           18 IF(NCS.EQ.0)  GO TO 20
0076              IF(NCSTR + NCS.LE.90)  GO TO 35
0077         1000 WRITE(6,1001)
0078         1001 FORMAT(38HONUMBER OF CONSTRAINTS HAS EXCEEDED 90)
0079              NCS = 90 - NCSTR
0080           35 DO 19 I = 1,NCS
0081              NCSTR = NCSTR + 1
0082              KODE(NCSTR) = KODX(I)
0083              CS(NCSTR) = CSX(I)
0084              NPROG(NCSTR) = NPRO(I)
0085              KPROG(NCSTR) = KPRO(I)
0086           19 CONTINUE
0087           20 CALL LISTC
0088              CALL PLOT1 (NSCALE,7,5,15,6)
0089              T = 1.0
0090              DO 17 I=1,20
0091              YEAR(I) = TREF + T - 1.
0092              Y(I) = AMOD(YEAR(I),100.)
0093           17 T = T + 1.0
0094              WRITE (6,903)
0095              NLV = 0
0096              DO 33 I = 1,NV
0097              DO 31 J = 1,M
0098              IF(IVEH(J).NE.I)   GO TO 31
0099              NLV = NLV + 1
0100              KVEH(I)  = NLV
0101              KVEHI(NLV) = I
0102              GO TO 32
0103           31 CONTINUE
0104              GO TO 33
0105           32 IA = VEH(1,I)
0106              IB = VEH(2,I)
0107              IC = VEH(3,I)
0108              ID = VEH(4,I)
0109              WRITE(6,905) I,STG(IA),STG(IB),STG(IC),STG(ID),RCOST(I)
0110           33 CONTINUE
0111              DO 335 I = 1,M
0112              NX = IVEH(I)
0113          335 CALL AFRMT (NX,VNAM(I))
0114              DO 39 PROG = 1,N
0115           39 CALL REVLUS
0116           22 DO 23 I=1,N
0117              NOP(I) = 0
0118              RF(I) = R(I)
0119              SF(I) = S(I)
0120              CF(I) = C(I)
0121              SUSTF(I) = SUS (I)
0122              NDSF(I) = NYRSST(I)
0123              NSSF(I) = NSTRST(I)
0124              NSXF(I) = NSTRFX(I)
0125              LNDF(I) = LNDATE(I)
0126           23 NSRF(I) = NSTRRC(I)
0127              DO 24 I=1,M
0128           24 LVSF(I) = LVS(I)
0129              IF(FINISH.GT.1)  GO TO 21
0130              DO 25 I=ISTRT,IFIN
0131           25 CNTRVL(I) = ASTR
            C  NOP = 1 IF NO CHANGES ARE ALLOWED IN PROGRAM VARIABLES
0132           21 DO 26 I = 1,NCSTR
0133              J = NPROG(I)
0134              IF (KODE(I).EQ.8) NOP(J) = 1
0135           26 CONTINUE
0136           27 OUT = .FALSE.
0137              DO 61 PROG = 1,N
0138              IF(NOP(PROG).EQ.1)  GO TO 61
0139              CALL CONSTR
0140              IF (IERR.NE.0) WRITE (6,91) PROG
0141           61 CONTINUE
0142           91 FORMAT('0WARNING - CONSTRAINT VIOLATED IN PROGRAM NUMBER',I3)
```

```
0143              DO 300 ITER = 1,MAXITR
0144              IPRNT = 0
0145              IF (ITER.EQ.MAXITR)  IPRNT = 1
0146              DO 200 PROG = 1,N
                C  IODD INDICATES WHAT TYPE OF CHANGE IS BEING MADE- IODD=0 INITIALLY
0147             14 SKIP = (IPRNT.EQ.0.AND.ITER.GT.1).OR.PROG.NE.1.OR.IODD.NE.0
0148              IF (SKIP.AND.NOP(PROG).EQ.1.AND.PROG.NE.1) GO TO 195
0149             15 DO 30 J=1,20
0150              TOTAL(J) = 0.
0151             30 W(J) = 0.
0152              IF (SKIP) GO TO 55
0153             40 XT = 0.
0154              ST = 0.
0155              DO 50 I=1,N
                C  FLAGR = * INDICATES A CHANGE IN DEVELOPMENT DURATION
0156              FLAGR(I) = BLANK
                C  FLAGS = * INDICATES A CHANGE IN START DATE OF DEVELOPMENT
0157              FLAGS(I) = BLANK
0158              IF (R(I).NE.RF(I)) FLAGR(I) = ASTR
0159              IF (S(I).NE.SF(I)) FLAGS(I) = ASTR
0160              X = NYRSST(I)
0161              ST = ST + SUS(I)*X
0162             50 XT = XT + C(I)
0163              WRITE (6,90) TREF,TITLE
0164              WRITE (6,92)
0165              DO 53 I=1,N
0166              IF(I.GT.NMIS+NSPR)   GO TO 52
0167              K = NYRSRC(I)
0168              IF(K.EQ.0.OR.I.GT.NMIS)  RECUR(1,I) = 0.0
0169              WRITE (6,94) I,NAME(I),S(I),FLAGS(I),C(I),R(I),FLAGR(I),SUS (I),
                 X  NSTRST(I),NYRSST(I),NSTRRC(I),NYRSRC(I),(RECUR(J,I),J=1,K)
0170              GO TO 51
0171             52 NDM = I - NMIS - NSPR
0172              WRITE(6,93) I,LABEL(NDM),S(I),FLAGS(I),C(I),R(I),FLAGR(I),SUS(I),
                 X  NSTRST(I),NYRSST(I),NSTRRC(I),NYRSRC(I)
0173             51 K = NYRSFX(I)
0174              IF (K.EQ.0) GO TO 53
0175              WRITE (6,98) NSTRFX(I),NYRSFX(I),(RFIXD(J,I),J=1,K)
0176             53 CONTINUE
0177              WRITE (6,95) XT,ST
0178              IF (ITER.NE.1) WRITE (6,902)
0179              WRITE(6,96) YEAR
0180              WRITE (6,97)

0181              CALL PLOT2 (IMAGE,Y(16),Y(1),PMAX,PMIN)
0182              DO 54 I=1,1000
0183             54 XLVSUM(I,1) = 0.0
                C
0184             55 CALL TCOSTS (BLANK,ASTR)
                C
0185              IF(LS.GT.20.OR.LS.LE.0)  GO TO 78
0186              XL = 0.
                C  IF LEVEL(1) AND (2) ARE INPUT AS INTEGER YEARS, THEN THE PROGRAM
                C  TAKES THE AVERAGE SPENDING OVER THE PERIOD ENCOMPASSED BY THESE
                C  YEARS AS THE DESIRED BUDGET LEVEL
0187              DO 76 I=LS,LF
0188             76 XL = XL+TOTAL(I)
0189              XL = XL/FLOAT(LF-LS+1)
0190              DO 77 I=1,20
0191             77 LEVEL(I) = XL
0192             78 IF (SKIP) GO TO 80
0193              WRITE (6,99) (W(I),I=1,JS)
0194              WRITE (6,990) (FIXED(I),I=1,JS)
0195              WRITE (6,991) (TOTAL(I),I=1,JS)
0196              WRITE (6,993) CNTRVL
0197              WRITE (6,992) (LEVEL(I),I=1,JS)
0198              IF(KSTAT.GT.0) WRITE(6,994) (XMODE(I),I = 1,JS)
0199              IF(KSTAT.GT.0)  WRITE(6,995) (UB(I), I = 1,JS)
0200              CALL PLOT3(FLET,Y,FIXED,JS)
0201              IF(KSTAT.GT.0)  CALL PLOT3(ULET,Y,UB,JS)
0202              IF(KSTAT.GT.0) CALL PLOT3(MLET,Y,XMODE,JS)
0203              CALL PLOT3 (ZERO,Y,LEVEL,IFIN)
0204              CALL PLOT3 (ASTR,Y,TOTAL,JS)
0205             80 SOD = 0
0206              DO 100 I=ISTRT,IFIN
0207              SOD = (TOTAL(I)-LEVEL(I))**2 + SOD
0208            100 CONTINUE
0209              RMS = SQRT (SOD/FLOAT(IFIN-ISTRT+1))
                C   SAVEX = RMS VALUE AT BEGINNING OF ITERATION
0210              IF (PROG.EQ.1.AND.IODD.EQ.0) SAVEX = RMS
0211              IF (SKIP) GO TO 110
                C  RMS1 = VALUE OF RMS USING INPUT DATA
0212              IF(ITER.EQ.1)  RMS1 = RMS
0213              WRITE (6,199) RMS,YEAR(ISTRT),YEAR(IFIN)
0214              WRITE (6,298) ITER
0215              WRITE (6,399)
0216              CALL PLOT4 (13,PRGLV)
```

```
0217              WRITE (6,499)
0218              IF(MOS.EQ.2.OR.MOS.EQ.3)  RETURN
0219            110 IF (OUT) GO TO 400
0220              IF (ITER.EQ.MAXITR) GO TO 300
                C  SAVER = RMS VALUE AT BEGINNING OF PROGRAM CHANGE CONSIDERATIONS
0221              IF(IODD.EQ.0)  SAVER = RMS
0222              IF(RMS.LT.SAVER)  GO TO 190
                C
0223              CALL SHIFTS
                C
0224              IF(MYFLAG.EQ.1)  GO TO 14
0225            190 IODD = 0
                C   SAVER  = VALUE OF RMS AT END OF ITERATION
0226              IF(RMS.LT.SAVER.AND.PROG.EQ.N)  SAVER = RMS
0227            195 IF(PROG.LT.N)   GO TO 200
0228              IF (SAVEX.NE.SAVER) GO TO 300
0229              IF (IPRNT.NE.0) GO TO 400
0230              SKIP = .FALSE.
0231              OUT = .TRUE.
0232              GO TO 15
0233            200 CONTINUE
0234            300 CONTINUE
0235              WRITE (6,390)
0236              GO TO 403
0237            400 WRITE (6,299)
0238            403 WRITE (6,906) (YEAR(I),I=1,JS)
0239              WRITE (6,907)
0240              DO 402 I=1,NLV
0241              XLVTOT = 0.0
0242              DO 401 II=1,JS
0243            401 XLVTOT = XLVTOT + XLVSUM(II,I)
                C  XLVSUM(II,I) = NUMBER OF LAUNCHES IN YEAR II FOR VEH. KVEHI(I)
0244            402 WRITE (6,908) KVEHI(I),XLVTOT,(XLVSUM(II,I),II=1,JS)
0245              IF(SAVER.LT.RMS1 - .4)  GO TO 404
0246              WRITE(6,909)
0247            909 FORMAT (46HOINPUT ASSIGNMENT IS OPTIMUM SMOOTHED SOLUTION)
0248              GO TO 7
0249            404 NNMI = NMIS + NSPR
0250              DO 9 I = 1,NNMI
0251              IF(ABS(S(I) + R(I) - SF(I) - RF(I)).GE..01) GO TO 13
0252              IF(NYRSST(I).NE.NDSF(I))  GO TO 13
0253              IF(NLVP(I).EQ.0)  GO TO 9
0254              IF(LNDATE(I).NE.LNDF(I))  GO TO 13
0255              IJ = NLVP(I)
0256              H = LVARY(I)
0257              DO 11 II = 1,IJ
0258              X = LVS(H) - LVSF(H)
0259              IF(ABS(S(I) + X - SF(I)).GE..01)  GO TO 13
0260             11 H = H + 1
0261              9 CONTINUE
0262              IF(N.EQ.NNMI)   GO TO 7
0263              NNMI = NNMI + 1
0264              DO 10 I = NNMI,N
0265              IF(ABS(S(I) + R(I) - SF(I) - RF(I)).GE.01)  GO TO 13
0266              IF(NYRSST(I).NE.NDSF(I))  GO TO 13
0267              IF(ABS(CF(I)-C(I)).GE..001)  GO TO 13
0268              IF(ABS(SUS(I) - SUSTF(I)).GE..001)  GO TO 13
0269             10 CONTINUE
0270              7 FINISH = MITR + 1
0271              GO TO 12
0272             13 FINISH = FINISH + 1
0273             12 NCSTR = NCSTR - NCS
0274              RETURN
0275             90 FORMAT (1H1,15X,14HREFERENCE YEAR,F7.0,5X,10A4)
0276             92 FORMAT (78HOPN  NAME  START    DEVL  YRS   SUST  SS  SD  RS  RD  R
                   XECURRING OR FIXED ITEMS   /1H )
0277             93 FORMAT (I3,1X,4HDEV ,I2,F6.0,1X,A1,F7.0,F4.0,1X,A1,F5.0,4I4)
0278             94 FORMAT (I3,1X,A6,F6.0,1X,A1,F7.0,F4.0,1X,A1,F5.0,4I4,12F6.0)
0279             95 FORMAT (20X,4H----,8X,4H----/2X,5HTOTAL,12X,F6.0,F11.0)
0280             96 FORMAT (1H1,30X,47HTOTAL PROGRAM COSTS AND LAUNCH VEHICLE SCHEDULE
                   *  /6HOYEAR ,4X,20F6.0)
0281             97 FORMAT (8HOPROGRAM)
0282             98 FORMAT(44X,2I4,12F6.0)
0283             99 FORMAT (6HOSUM  ,4X,20F6.0)
0284            107 FORMAT (44X,2I4,12F6.1)
0285            199 FORMAT (6HORMS =,F8.0,5X,18HSMOOTHING INTERVAL,F6.0,5H THRU,F6.0)
0286            298 FORMAT (1OHOITERATION, I3)
0287            299 FORMAT (11X,11H FINAL CASE)
0288            390 FORMAT (11X,16H MAXITR EXCEEDED )
0289            399 FORMAT (1H1)
0290            499 FORMAT (1HO,50X,4HYEAR)
0291            902 FORMAT (1HO,40X,34H* INDICATES CHANGE FROM INPUT DATA)
0292            903 FORMAT (1H1,30X,19HRECURRING COST DATA  /1HO,8X,3HKEY,10X,
                   *  4HNAME,24X,9HUNIT COST  /1H )
0293            905 FORMAT (10X,I2,10X,4A4,10X,F10.2)
0294            906 FORMAT (1H1,30X,35HLAUNCH VEHICLE REQUIREMENTS BY YEAR  /
```


40

```
FORTRAN IV G LEVEL 1, MOD 4            SMOOTH            DATE = 71312          18/12/29

                     *  6HOYEAR ,6X,20F6.0)
0295                907 FORMAT (11HOLV    TOTAL)
0296                908 FORMAT (1X,I2,F8.2,20F6.1)
0297                990 FORMAT (6H FIXED,4X,20F6.0)
0298                991 FORMAT (6H TOTAL,4X,20F6.0)
0299                992 FORMAT (6H LEVEL,4X,20F6.0)
0300                993 FORMAT (8X,20(5X,A1))
0301                994 FORMAT(6HO MODE,4X,20F6.0)
0302                995 FORMAT(12H 50 PER CENT/8H CONFID.,2X,20F6.0)
0303                    END


TOTAL MEMORY REQUIREMENTS 0022AC BYTES


F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000      NAME MOX02SS(R)
IEW0461  IBCOM=
IEW0461  INPUT
IEW0461  FRXPI=
IEW0461  LISTC
IEW0461  PLOT1
IEW0461  AFRMT
IEW0461  REVLUS
IEW0461  CONSTR
IEW0461  PLOT2
IEW0461  TCOSTS
IEW0461  PLOT3
IEW0461  PLOT4
IEW0461  SHIFTS
IEW0461  SQRT

                                        MODULE MAP

  CONTROL SECTION                    ENTRY
    NAME     ORIGIN   LENGTH          NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

  SMOOTH        00     22AC
  SAVER       22B0      FC0
  SAVE1       3270      FC4
  SAV2        4238      FE0
  SAV3        5218      980
  SAVALL      5B98     3A1C
  VARNCE      95B8      ADC
  SCRACH      A098     6A60

  ENTRY ADDRESS       00
  TOTAL LENGTH     10AF8

****MOX02SS   NOW REPLACED IN DATA SET
```

```
0001            SUBROUTINE STGNMI
              C DETERMINE NUMBER OF COMPONENTS ACTUALLY USED AND ASSOCIATED
              C   RECURRING COSTS
              C
0002            REAL NPERPD
0003            INTEGER*2  NU,NBY,MODE,NOB,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,
                1  LABF,LABI,NPSTG,NPAD,NPFAM,NFS,NPINTL,NPINTU,MAPS,MAPF,MAPI,VEH,
                2  NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,KOUT,LTR,NINTYR,NTGYTR,
                3   MAF,MAIC
0004            COMMON/SAVSAR/COR,POJ(3),SRJ(3,3),NU(40),NBY(40),NOB(40),RINT(40),
                1   PLCINT(40),XLT(40), PLCT(40),UPP(40),TAT(40),TAMT(50),SR(40,3),
                2   MODE(40,3),PLC(40,3)
0005            COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                2  RPLM(50),MAS(40,3), RXD(12,50)
0006            COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0007            COMMON/SAV4/  MAF(30,3), MAIC(40,3),
                *             NPAD(2,60),NPFAM(30,5),NPINTL(30,5),NPINTU(30,5),
                1  NFS(40,4),NPSTG(30,10),MAPS(30,10),MAPF(30,10),MAPI(30,10),
                2  PFAMD(30,5,2),PFAMS(30,5,2),PINTS(30,5,2),PSTGD(30,10,2),
                3   PSTGS(30,10,2)
0008            COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0009            COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
                1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
0010            COMMON/SCRACH/  II,NUS(40),MSAVE(40),ISAVE(40),KLUE(40),
                1STGYHW(40,20),RINTMX(40,20),STGMAX(40,20,2),STGYTR(40,20,2),
                2  RINTYR(40,20),DUM(1047)
              C
0011            IF (IFLAG.GE.1) GO TO 621
              C     FIND MAX NUM OF EACH STAGE AND INTEGRATION POSSIBLE
0012            DO 661 I = 1,NSTG
0013            NUS(I) = 0
0014            DO 661 J=1,MYRS
0015            STGYTR(I,J,1) = 0.0
0016          661 STGYTR(I,J,2) = 0.0
0017            IF(NCI.EQ.0)  GO TO 665
0018            DO 662 I = 1,NCI
0019            DO 662 J = 1,MYRS
0020          662 RINTYR(I,J) = 0.0
0021          665 DO 6500  I = 1,NM
0022            IF(YRLM(I).LT..001)  GO TO 6500
0023            J = LYR(I)
0024            JX = LETT(I)
0025            K = LTR(JX)
              C   MSAVE & ISAVE INDICATE IF THAT STAGE OR INTEGRATION HAS ALREADY BEEN
              C     COUNTED FOR MISSION NM
0026            DO 9003 IZ = 1,40
0027            MSAVE(IZ) = 0
0028         9003 ISAVE(IZ) = 0
0029            DO 650 I1 = 1,NV
0030            IF(ITEM(VNM(1,I),I1,1).EQ.0)  GO TO 650
0031            X = NMULT(I1,JX)
0032            DO 649 MS = 1,4
0033            IA = VEH(MS,I1)
0034            IF(IA.EQ.0)  GO TO 650
0035            IF(MSAVE(IA).EQ.1)  GO TO 644
0036            STGYTR(IA,J,K) = YRLM(I)*X + STGYTR(IA,J,K)
0037            MSAVE(IA) = 1
0038          644 IF(NCI.EQ.0)  GO TO 649
0039            IF(MS.EQ.4)  GO TO 650
0040            IF(VEH(MS+1,I1).EQ.0)  GO TO 650
0041            L1 = VEH(MS+1,I1)
0042            DO 645 MI = 1,NCI
0043            IF(ISAVE(MI).EQ.1)  GO TO 645
0044            DO 646 KY = 1,4
0045            IF(NFML(MI).NE.NFS(IA,KY))  GO TO 646
0046            DO 647 KZ = 1,4
0047            IF(NFMU(MI).EQ.NFS(L1,KZ))   GO TO 648
0048          647 CONTINUE
0049          646 CONTINUE
0050            GO TO 645
0051          648 RINTYR(MI,J) = RINTYR(MI,J) + YRLM(I)*X
0052            ISAVE(MI) = 1
0053          645 CONTINUE
0054          649 CONTINUE
0055          650 CONTINUE
0056         6500 CONTINUE
0057            DO 668 J = 1,MYRS
0058            DO 668 I = 1,NSTG
0059          668 STGYHW(I,J) = STGYTR(I,J,1) + STGYTR(I,J,2)
0060            GO TO 673
              C
              C DETERMINE NUMBER OF EACH STAGE AND INTEGRATION USED IN LAST ITERATION BY YEAR
```

```
                C              FOR FUTURE PRINT OUT
0061              621 DO 623 K=1,2
0062                  DO 623 J=1,MYRS
0063                  DO 623 I=1,NSTG
0064                  STGMAX(I,J,K) = NTGYTR(I,J,K)
0065                  STGMAX(I,J,K) = STGMAX(I,J,K)/10.0
0066              623 STGYTR(I,J,K) = 0.0
0067                  IF(NCI.EQ.0)  GO TO 9000
0068                  DO 624 J=1,MYRS
0069                  DO 624 I=1,NCI
0070                  RINTMX(I,J) = NINTYR(I,J)
0071                  RINTMX(I,J) = RINTMX(I,J)/10.0
0072              624 RINTYR(I,J) = 0.0
0073             9000 DO 622 J=1,NM
0074                  IF(YRLM(J).LT..001)  GO TO 622
0075                  I = MIN(J)
0076                  K = LYR(J)
0077                  JX = LETT(J)
0078                  ITR = LTR(JX)
0079                  X = NMULT(I,JX)
0080                  DO 625 MS = 1,4
0081                  L = VEH(MS,I)
0082                  IF (L.EQ.0) GO TO 622
0083                  STGYTR(L,K,ITR) = STGYTR(L,K,ITR) + YRLM(J)*X
0084                  IF (NCI.EQ.0) GO TO 625
0085                  IF (MS.EQ.4) GO TO 622
0086                  IF (VEH(MS+1,I).EQ.0) GO TO 622
0087                  L1 = VEH(MS+1,I)
0088                  DO 626 MI=1,NCI
0089                  DO 627 KY=1,4
0090                  IF(NFML(MI).NE.NFS(L,KY)) GO TO 627
0091                  DO 628 KZ = 1,4
0092                  IF (NFMU(MI).EQ.NFS(L1,KZ)) GO TO 629
0093              628 CONTINUE
0094              627 CONTINUE
0095                  GO TO 626
0096              629 RINTYR(MI,K) = RINTYR(MI,K) + YRLM(J)*X
0097              626 CONTINUE
0098              625 CONTINUE
0099              622 CONTINUE
0100                  IF(NCI.EQ.0)  GO TO 9001
0101                  DO 691 I=1,NCI
0102                  DO 691 J=1,MYRS
0103              691 IF(RINTYR(I,J).LT..001)  RINTYR(I,J) = RINTMX(I,J)
                C  CHECK NUMBER OF LAUNCHES CALCULATED VS. NUMBER OF LAUNCHES USED IN LAST
                C                 ITERATION
0104             9001 IF(LCK.EQ.0.OR.MOS.EQ.1.OR.MOS.EQ.3)    GO TO 4100
0105                  DO 676 K = 1,2
0106                  DO 676 J = 1,MYRS
0107                  DO 676 I = 1,NSTG
0108                  IF(ABS(STGYTR(I,J,K) - STGMAX(I,J,K)).GT.0.001.AND.STGYTR(I,J,K).
                     1GT.0.001.OR.(IFLAG.LE.1.AND.NU(I).LT.0))   GO TO 677
0109              676 CONTINUE
0110             4100 WRITE(6,4101)
0111             4101 FORMAT (1HO,4X, 40HTHE OPTIMUM SOLUTION HAS BEEN DETERMINED)
                C
0112              678 CALL VEHRC
                C
0113                  IFLAG = 0
0114                  IF(LOUT.EQ.0)  RETURN
0115                  DO 112 I = 1,NUMD
0116                  IF(KOUT(I).EQ.0)  GO TO 112
0117                  LT = KOUT(I)
0118                  SUST(I) = SAVS(LT)
0119              112 CONTINUE
0120                  RETURN
                C
0121              677 IF(IFLAG.LE.3)  GO TO 679
0122                  WRITE(6,8005)
0123             8005 FORMAT(49HOMAXIMUM NUMBER OF ASSIGNMENT ITERATIONS EXCEEDED)
0124                  GO TO 678
                C     DETERMINE HARDWARE COSTS BY YEAR BASED ON LAST ITERATION
0125              679 DO 8013 I = 1,NSTG
0126                  KLUE(I) = 0
0127                  DO 8014 J = 1,MYRS
0128                  IF(STGYTR(I,J,1).GT.0.01.OR.STGYTR(I,J,2).GT..01)  KLUE(I) = 1
0129             8014 STGYHW(I,J) = STGYTR(I,J,1) + STGYTR(I,J,2)
0130                  IF(KLUE(I).EQ.1)  GO TO 8013
0131                  DO 8016 J = 1,MYRS
0132                  STGYTR(I,J,1) = STGMAX(I,J,1)
0133                  STGYTR(I,J,2) = STGMAX(I,J,2)
0134             8016 STGYHW(I,J) = STGYTR(I,J,1) + STGYTR(I,J,2)
0135             8013 CONTINUE
                C
                C   ADD INITIAL REUSABLE PURCHASE PRICE TO DEV. COST DS
0136              672 IF(IFLAG.GT.1.OR.FINISH.GT.1)  GO TO 673
```

```
0137                NX = NSTG + 1
0138                DO 674 I1 = 1,NX
0139                READ(5,5000) J,NX2, X3,X4,X5
0140                IF(J.EQ.0)  GO TO 6735
0141                IF(I1.EQ.1)  WRITE(6,5003)
0142                DO 675 I = 1,NSTG
0143                IF(J.EQ.KODS(I))  GO TO 6755
0144             675 CONTINUE
0145                WRITE(6,5001) I1
0146                IFLAG = 100
0147                GO TO 674
0148            6755 NOB(I) = NX2
0149                XLT(I) = X3
0150                TAT(I) = X4
0151                IF(X5.GT..001)  PLCT(I) = ALOG(X5)/ALOG(2.)
0152                IF(X5.LE..001)  PLCT(I) = 0.0
0153                WRITE(6,5002) J,I,NU(I),UPP(I),NOB(I),XLT(I),TAT(I),PLCT(I)
0154             674 CONTINUE
0155            6735 IF(IFLAG.EQ.100) RETURN
0156             673 K = 0
0157                DO 710 I = 1,NSTG
0158                IF((NU(I).EQ.0).OR.(IFLAG.GE.1.AND.NU(I).GE.0))  GO TO 710
0159                IF(IFLAG.EQ.0)  GO TO 709
0160                NUS(I) = NU(I)
0161                II = I
                 C      ITERATES ON INITIAL QUANTITY TO BE PURCHASED
                 C
0162                CALL REUSE
                 C
0163             709 NI = MAS(I,1)
0164                IF(NU(I).LT.0)  LCK = 1
0165                X = NU(I)
0166                IF(X.LT.0)  X = -X
0167                Y = NUS(I)
0168                DS(NI) = DS(NI) + (X+Y) * UPP(I)
0169                IF(K.EQ.0)  WRITE (6,211)
0170                K = 1
0171                WRITE(6,208) NI,DS(NI),STS(I),STG(I),NYD(NI),LYD(NI)
0172                WRITE (6,209)  X
0173             710 CONTINUE
                 C      MAKE ADJUSTMENT FOR BATCHING OVER YEARS
0174                DO 663 I = 1,NSTG
0175                IF (NBY(I).EQ.1) GO TO 663
0176                IA = 2
0177                IB = NBY(I)
0178                IC = 1
0179             666 DO 664 J= IA,IB
0180                IF (J.GT.MYRS) GO TO 700
0181             664 STGYHW(I,IC) = STGYHW(I,IC) + STGYHW(I,J)
0182             700 DO 667 J = IA,IB
0183                IF (J.GT.MYRS) GO TO 663
0184             667 STGYHW(I,J) = STGYHW(I,IC)
0185                IA = IA + NBY(I)
0186                IB = IB + NBY(I)
0187                IC = IC + NBY(I)
0188                GO TO 666
0189             663 CONTINUE
0190                IF(IFLAG.EQ.0)  GO TO 9006
0191                DO 9005 I = 1,NSTG
0192                IF(KLUE(I).EQ.0)  GO TO 9005
0193                DO 9002 J = 1,MYRS
0194                IF(STGYTR(I,J,1).LT..01) STGYTR(I,J,1) = STGMAX(I,J,1)
0195                IF (STGYTR(I,J,2).LT..01) STGYTR(I,J,2) = STGMAX(I,J,2)
0196                IF(STGYHW(I,J).LT..01)  STGYHW(I,J) = STGYTR(I,J,1)+STGYTR(I,J,2)
0197             9002 CONTINUE
0198             9005 CONTINUE
0199             9006 DO 9007 K = 1,2
0200                DO 9007 J = 1,MYRS
0201                DO 9007 I = 1,NSTG
0202             9007 NTGYTR(I,J,K) = STGYTR(I,J,K)*10.0
0203                IF(NCI.EQ.0)  GO TO 9009
0204                DO 9008  J = 1,MYRS
0205                DO 9008 I = 1,NCI
0206             9008 NINTYR(I,J) = RINTYR(I,J)*10.0
                 C
                 C  DETERMINE VEHICLE RECURRING COSTS BY YEAR AND LAUNCH SITE
0207             9009 DO 632 I=1,NV
0208                DO 635 J=1,MYRS
0209                RECUR(I,J,1) = 0.0
0210             635 RECUR(I,J,2) = 0.0
0211                DO 633 MS = 1,4
0212                K = VEH(MS,I)
0213                IF (K.EQ.0) GO TO 632
0214             9004 DO 634 J= 1,MYRS
0215                IF(STGYHW(K,J).LT.0.001)  GO TO 634
0216                IF(MODE(K,1).NE.0)  GO TO 8015
```

```
0217              HDWR = SR(K,1)*STGYHW(K,J)**PLC(K,1)
0218              GO TO 8010
0219            8015 LX = MODE(K,1)
0220              IF(STGYHW(K,J).LE.POJ(LX))   HDWR = SRJ(LX,1)/STGYHW(K,J)
0221              IF(STGYHW(K,J).GT.POJ(LX))  HDWR = SRJ(LX,2)+SRJ(LX,3)/STGYHW(K,J)
0222            8010 DO 692 L = 1,2
0223              IF(STGYTR(K,J,L).LT..001)   GO TO 692
0224              M = L + 1
0225              IF(MODE(K,M).NE.0)  GO TO 8011
0226              RECUR(I,J,L)=RECUR(I,J,L)+HDWR+SR(K,M)*STGYTR(K,J,L)**PLC(K,M)
0227              GO TO 692
0228            8011 LX = MODE(K,M)
0229              IF(STGYTR(K,J ,L).LE.POJ(LX)) RECUR(I,J ,L) = RECUR(I,J ,L)
                 1  + SRJ(LX,1)/STGYTR(K,J ,L) + HDWR
0230              IF (STGYTR(K,J ,L).GT.POJ(LX)) RECUR(I,J ,L) = RECUR(I,J ,L) +
                 1          SRJ(LX,2) + SRJ(LX,3)/STGYTR(K,J ,L) + HDWR
0231             692 CONTINUE
0232             634 CONTINUE
0233              IF(NCI.EQ.0)  GO TO 633
0234              IF (MS.EQ.4) GO TO 632
0235              IF(VEH(MS+1,I).EQ.0) GO TO 632
0236              K1 = VEH(MS+1,I)
0237              DO 636 L=1,NCI
0238              DO 637 KY=1,4
0239              IF (NFML(L).NE.NFS(K,KY)) GO TO 637
0240              DO 638 KZ = 1,4
0241              IF (NFMU(L).EQ.NFS(K1,KZ)) GO TO 639
0242             638 CONTINUE
0243             637 CONTINUE
0244              GO TO 636
0245             639 DO 640 J = 1,MYRS
0246              IF(RINTYR(L,J).LT..0001)  GO TO 640
0247              HDWR = RINT(L)*RINTYR(L,J)**PLCINT(L)
0248              RECUR(I,J,1) = RECUR(I,J,1) + HDWR
0249              RECUR(I,J,2) = RECUR(I,J,2) + HDWR
0250             640 CONTINUE
0251             636 CONTINUE
0252             633 CONTINUE
0253             632 CONTINUE
0254              99 RETURN
0255             208 FORMAT (I4,6X,2F12.2,5X,A4,1X,14HSTAGE HARDWARE,29X,I3,9X,I3)
0256             209 FORMAT (6X, 27HNUMBER OF UNITS PURCHASED =,  F5.1)
0257             211 FORMAT (33H1CHANGED QUANTITIES BRANCHED UPON/1HO,6HNUMBER, 5X,

                1  11HDEVELOPMENT, 2X, 10HSUSTAINING, 50X, 10HYEAR AVAIL, 2X,
                2  9HLAST YEAR//)
0258            5000 FORMAT (I2,2X,I2,2F6.1,F6.3)
0259            5001 FORMAT (45HOKODE NUMBER INCORRECT ON REUSABLE STAGE CARD, I6)
0260            5002 FORMAT (I4,I6, I8, F10.1, I6, F11.0, F10.0, F8.2)
0261            5003 FORMAT (22H1  REUSABLE STAGE DATA//5H KODE,3X,5HORDER,3X,5HUNITS,
                2   3X,5HPRICE,3X,4HTYPE,3X,8HLIFETIME,3X,7HTA TIME,3X,2HLC)
0262              END
```

TOTAL MEMORY REQUIREMENTS 002314 BYTES

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000       NAME MOX02SM(R)
IEW0461   ITEM
IEW0461   IBCOM=
IEW0461   VEHRC
IEW0461   REUSE
IEW0461   FRXPR=
IEW0461   ALOG
```

MODULE MAP

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
STGNMI	00	2314						
SAVSAR	2318	A5C						
SAVE1	2D78	FC4						
SAV3	3D40	980						
SAV4	46C0	3188						
SAVALL	7848	3A1C						
TEMP	B268	4110						
SCRACH	F378	6A60						

ENTRY ADDRESS 00
TOTAL LENGTH 15DD8

****MOX02SM NOW REPLACED IN DATA SET

```
FORTRAN IV G LEVEL 1, MOD 4          TCOSTS          DATE = 71312          16/52/59

0001                 SUBROUTINE TCOSTS (BLANK,ASTR)
                C    CALCULATE TOTAL COSTS
                C
0002                 DOUBLE PRECISION NAME
0003                 LOGICAL SKIP,EXT,ACCL
0004                 REAL NPERPD
0005                 INTEGER H,PROG
0006                 INTEGER*2  LTR,YDPL,NSYR,NSFX,NRFX,NYRSST,NSTRFX,NPROG,KPROG,KODE,
                    1  NYRSFX,KODEM,KODESP,VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN
                    2,FINISH,NSTG,NFML,NFMU,KODS,MAS,LABS,LABF,LABI,
                    6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
0007                 COMMON/SAVRT/RVAR(20,50)
0008                 COMMON/VARNCE/KSTAT,VARI(40),VARF(50),VARM(56),FMVAR(2,30),
                    1  FIVAR(3,40),PLVAR(3,56),SVAR(5,40)
0009                 COMMON/SAVE1/ FINISH,NSTG,NCI,ILY,LABF(30),LABS(40),LABI(40),
                    1 NFML(40),NFMU(40),KODS(40),STS(41),STG(40),VLR(50),WPR(50),
                    2  RPLM(50),MAS(40,3), RXD(12,50)
0010                 COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
                    1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                    2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
0011                 COMMON/SAVER/  RFIXD(12,84)
0012                 COMMON/SAV2/EXT,ACCL,KNSTG,KNFAM,KNCI,KNP,KNMIS,JFLAG,TREF,NCSTR,
                    1 PMAX,PMIN,ISTRT,IFIN,MAXITR,MITR,KODESP(6),TITLE(10),LEVEL(20),
                    2  CNTRVL(20),FIXED(20),KODEM(50),NSYR(50),NSFX(50),NAME(56),
                    3  YDPL(56),NRFX(50),NYRSST(84),NSTRFX(84),NYRSFX(84),SUS(84),C(84)
                    4,  R(84),  S(84),CS(90),NPROG(90),KPROG(90),  KODE(90)
0013                 COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
                    1    PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
0014                 COMMON/SCRACH/M,N,NCS,PROG,IODD,IERR,SKIP,MYFLAG,JS,NSCALE(5),
                    1  NSL(10),TOTAL(20),W(20),D(20),XOUT(20),VOUT(20),RRR(20),YEAR(20)
                    2,  Y(20),KVEHI(50),LABEL(50),LVARY(70),LVD(70),IVEH(70),LVS(70),
                    3  LVSF(80),VNAM(80),NOP(86),RF(86),CF(86),SF(86),FLAGR(86),
                    4  FLAGS(86),NSSF(86),NSRF(86),NSXF(86),NDSF(86),SUSTF(86),NLVP(86)
                    5,  NSTRRC(86),NYRSRC(86),LNDF(86),NSTRST(86),LNDATE(86),NPRO(90),
                    6  KPRO(90),CSX(90),LZ(46),RCOST( 60),  KVEH( 60),IMAGE(830),
                    7  XSCH(10,70),PLSCH(10,70),XLVSUM(20,50),RECUR(20,50),DUM(340),
                    8  VTC(20),XMODE(20),U8(20),DUMMY(1)
                C
0015                 DO 55 I = 1,20
0016                55 VTC(I) = 0.0
0017                 JS= 0
0018                 DO 70 L=1,N
0019                 FLAG = 0.
```

```
0020              T = 1.0
0021              AYRS = R(L) + 1.0
0022              IF(L.LE.NMIS + NSPR) GO TO 30
0023              NDUM = L - NMIS - NSPR
0024              LX = LABEL(NDUM)
0025              JX = MAT(LX)
0026              IF(JX.GT.1000) JX = JX - 2000
0027             30 DO 60 K=1,20
0028                F = 0.
0029                IT = T - S(L) + TREF
0030                X = (T - S(L) + TREF) / AYRS
                 C  X.LE.0 PROGRAM DEV. HASN'T STARTED YET - X.GE.1 PROGRAM DEV. IS OVER
0031                IF (X.LE.0.) GO TO 59
0032                IF (X.GE.1.) GO TO 56
                 C  BETA DISTRIBUTION FOR C(L)
0033                F = ((X*(1.-X))**2) * 30. * C(L) / AYRS
0034                IF(KSTAT.EQ.0.OR.SKIP.OR.F.LT..0001)  GO TO 56
0035                IF(L.LE.NMIS + NSPR) GO TO 31
0036                IF(JX.LT.-200) GO TO 56
0037                IF(JX.LT.-100)  GO TO 41
0038                IF(JX.LT.0)  GO TO 42
0039                A = F*EXP(1.5*SVAR(4,JX))/(1.0 + EXP(1.5*SVAR(4,JX)))
0040                IF(SVAR(4,JX).GT..001)  VTC(K) = VTC(K) + A*A*(EXP(SVAR(4,JX)) -
                   1    1.0)
0041                GO TO 56
0042             41 KX = -JX - 100
0043                A = F*EXP(1.5*FIVAR(2,KX))/(1.0 + EXP(1.5*FIVAR(2,KX)))
0044                IF(FIVAR(2,KX).GT..001)                          VTC(K) = VTC(K) +
                   1    A*A*(EXP(FIVAR(2,KX      )) - 1.0)
0045                GO TO 56
0046             42 KX = - JX
0047                A = F*EXP(1.5*FMVAR(1,KX))/(1.0 + EXP(1.5*FMVAR(1,KX)))
0048                            IF(FMVAR(1, KX).GT..001)  VTC(K) = VTC(K) + A*A*
                   1  (EXP(FMVAR(1, KX)) - 1.0)
0049                GO TO 56
0050             31 IF(PLVAR(2,L).LT..0001)  GO TO 56
0051                A = F*EXP(1.5* PLVAR(2,L))/(1.0 + EXP(1.5*PLVAR(2,L)))
0052                VTC(K) = VTC(K) + A*A*(EXP(PLVAR(2,L)) - 1.0)
0053             56 IF (NYRSST(L).EQ.0) GO TO 57
0054                I = IT - NSTRST(L)
0055                IF(I.LT.0.OR.I.GE.NYRSST(L))  GO TO 57
0056                F = F + SUS(L)
0057                IF(KSTAT.EQ.0.OR.SKIP.OR.SUS(L).LT..0001)  GO TO 57
0058                IF(L.LE.NMIS + NSPR)  GO TO 32
0059                IF(JX.LT.-200)  GO TO 57
0060                IF(JX.LT.-100)  GO TO 43
0061                IF(JX.LT.0)  GO TO 44
0062                A = SUS(L)*EXP(1.5*SVAR(5,JX))/(1.0 + EXP(1.5*SVAR(5,JX)))
0063                IF(SVAR(5,JX).GT..001)  VTC(K) = VTC(K) + A*A*(EXP(SVAR(5,JX))-1.)
0064                GO TO 57
0065             43 KX = -JX -100
0066                A = SUS(L)*EXP(1.5*FIVAR(3,KX))/(1.0 + EXP(1.5*FIVAR(3,KX)))
0067                            IF(FIVAR(3, KX    ).GT..001)  VTC(K) = VTC(K) +
                   1  A*A*(EXP(FIVAR(3,KX)) - 1.0)
0068                GO TO 57
0069             44 KX = -JX
0070                A = SUS(L)*EXP(1.5*FMVAR(2,KX))/(1.0 + EXP(1.5*FMVAR(2,KX)))
0071                            IF(FMVAR(2, KX).GT..001)  VTC(K) = VTC(K) +
                   1            A*A*(EXP(FMVAR(2, KX)) - 1.0)
0072                GO TO 57
0073             32 IF(PLVAR(3,L).LT..0001)  GO TO 57
0074                A = SUS(L)*EXP(1.5*PLVAR(3,L ))/(1.0 + EXP(1.5*PLVAR(3,L)))
0075                VTC(K) = VTC(K) + A*A*(EXP(PLVAR(3,L)) - 1.0)
0076             57 IF (NYRSRC(L).EQ.0) GO TO 58
0077                I = IT - NSTRRC(L)
0078                IF(I.LT.0.OR.I.GE.NYRSRC(L))  GO TO 58
0079                F = F + RECUR(I+1,L)
0080                IF(KSTAT.EQ.0.OR.SKIP.OR.RECUR(I+1,L).LT..01)    GO TO 58
0081                A = RECUR(I+1,L)*EXP(1.5*RVAR(I+1,L))/(1.0+EXP(1.5*RVAR(I+1,L)))
0082                VTC(K) = VTC(K) + A*A*(EXP(RVAR(I+1,L)) - 1.0)
0083             58 IF (NYRSFX(L).EQ.0) GO TO 59
0084                I = IT - NSTRFX(L)
0085                IF(I.LT.0.OR.I.GE.NYRSFX(L))  GO TO 59
0086                F = F + RFIXD(I+1,L)
0087                IF(KSTAT.EQ.0.OR.SKIP.OR.RFIXD(I+1,L).LT..0001)  GO TO 59
0088                IF(L.LE.NMIS + NSPR)  GO TO 39
0089                IF(JX.LT.-200)  GO TO 59
0090                IF(JX.LT.-100)  LXX = LABI(-JX - 100)
0091                IF(JX.LT.0.AND.JX.GE.-100)  LXX = LABF(-JX)
0092                IF(JX.GT.0)  LXX = LABS(JX)
0093                A = RFIXD(I+1,L)*EXP(1.5*VARF(LXX))/(1.0 + EXP(1.5*VARF(LXX)))
0094                IF(VARF(LXX).GT..001)  VTC(K) = VTC(K) + A*A*(EXP(VARF(LXX)) -1.0)
0095                GO TO 59
0096             39 A = RFIXD(I+1,L)*EXP(1.5*VARM(L))/(1.0 + EXP(1.5*VARM(L)))
0097                IF(VARM(L).GT..001)  VTC(K) = VTC(K) + A*A*(EXP(VARM(L)) - 1.0)
0098             59 D(K) = F*(GRO + 1.)**(K-1)
```

```
0099                C   W(K) IS TOTAL COST IN YEAR K
                        W(K) = W(K) + D(K)
0100                    IF (D(K).EQ.0..AND.FLAG.EQ.1.) GO TO 65
0101                    IF (D(K).NE.0.) FLAG = 1.
0102                 60 T = T + 1.0
0103                    K = 21
0104                 65 K = K-1
0105                    JS = MAXO (JS,K)
0106                    IF (SKIP) GO TO 70
0107                    IF(L.LE.NMIS+NSPR)
                       1WRITE (6,98)  L,NAME(L),(D(I),I=1,K)
0108                    IF(L.GT.NMIS+NSPR)  WRITE(6,89) L,LABEL(NDUM),(D(I),I=1,K)
0109                    IF (NLVP(L).EQ.0) GO TO 70
0110                    IJ = NLVP(L)
0111                    H = LVARY(L)
0112                    DO 69 II=1,IJ
0113                    DO 67 I=1,20
0114                    XOUT(I) = BLANK
0115                 67 VOUT(I) = BLANK
0116                    XSUB = LVS(H)
0117                    IA = S(L) - TREF + XSUB
0118                    IB = IA+LVD(H)-1
0119                    DO 68 I=IA,IB
0120                    IF(I.LT.1.OR.I.GT.20)  GO TO 68
0121                    IC = I-IA+1
0122                    IF(XSCH(IC,H).LT..01)  GO TO 68
0123                    XOUT(I) = ASTR
0124                    VOUT(I) = VNAM(H)
0125                    KK = IVEH(H)
0126                    ILV = KVEH(KK)
0127                    XLVSUM(I,ILV) = XLVSUM(I,ILV) + XSCH(IC,H)
0128                 68 CONTINUE
0129                    WRITE (6,901) (VOUT(I),XOUT(I),I=1,K)
0130                 69 H = H + 1
0131                 70 CONTINUE
0132                    DO 75 I=1,JS
0133                    VTC(I) = VTC(I)*(1.0 + GRO)**(2*(I-1))
0134                 75 TOTAL(I) = W(I) + FIXED(I)
0135                    IF(KSTAT.EQ.0.OR.SKIP)  RETURN
0136                    DO 80 I = 1,JS
0137                    IF(W(I).LT..0001)  GO TO 79
0138                    IF(VTC(I).LT..01)  GO TO 78
0139                    A = .6*W(I)
0140                    SIG = ALOG(A*A + VTC(I)) - ALOG(A*A)
0141                    TP = SQRT(SIG)
0142                    XMODE(I) =    A*(EXP(-1.5*SIG))
0143                    XMU = ALOG(A)    - .5*SIG
0144                    XMD = (ALOG(XMODE(I)) - XMU)/TP
0145                    CALL NDTR(XMD,P2,DD)
0146                    P2 = P2 + .5
0147                    CALL NDTRI(P2,Y2,DD,IE)
0148                    UB(I) = EXP(TP*Y2 + XMU) + FIXED(I) + .4*W(I)
0149                    XMODE(I) = XMODE(I) + FIXED(I) + .4*W(I)
0150                    GO TO 80
0151                 78 XMODE(I) = TOTAL(I)
0152                    UB(I) = TOTAL(I)
0153                    GO TO 80
0154                 79 XMODE(I) = 0.0
0155                    UB(I) = 0.0
0156                 80 CONTINUE
0157                    RETURN
0158                 89 FORMAT(I3,1X,4HDEV ,I2,20F6.0)
0159                 98 FORMAT (I3,1X,A6,20F6.0)
0160                901 FORMAT (11X,20(A4,A2))
0161                    END
```

TOTAL MEMORY REQUIREMENTS 001A46 BYTES

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,NCAL,MAP
          VARIABLE OPTIONS USED - SIZE=(126976,24576)                    DEFAULT OPTION(S) USED
IEW0000       NAME MOXO2TC(R)
IEW0461  FRXPI=
IEW0461  IBCOM=
IEW0461  NDTR
IEW0461  NDTRI
IEW0461  EXP
IEW0461  MAXO
IEW0461  ALOG
IEW0461  SQRT

                                                  MODULE MAP

  CONTROL SECTION                          ENTRY
    NAME     ORIGIN   LENGTH                NAME   LOCATION     NAME   LOCATION     NAME   LOCATION

  TCOSTS        00     1A46
  SAVRT       1A48      FA0
  VARNCE      29E8      ADC
  SAVE1       34C8      FC4
  SAVALL      4490     3A1C
  SAVER       7EB0      FC0
  SAV2        8E70      FE0
  SAV3        9E50      980
  SCRACH      A7D0     6A60

 ENTRY ADDRESS         00
 TOTAL LENGTH       11230

****MOXO2TC   NOW REPLACED IN DATA SET
```

```
(17)          OS/360  FORTRAN H                                   DATE  71.312/18.25.49

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=44,SOURCE,BCD,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,
ISN 0002              SUBROUTINE VEHRC
                    C     DETERMINE 'AVERAGE' RECURRING COST OF EACH VEHICLE
                    C
                          REAL NPERPD
ISN 0003                  INTEGER*2  VEH,NMULT,NONREC,NYD,IS,MAT,LYR,LETT,LYD,MIN,KOUT,
ISN 0004                 1  NINTYR,NTGYTR,LTR,
                         6  KVEHI,LABEL,LVARY,LVD,IVEH,LVS,LVSF,NOP,NSSF,NSRF,NSXF,NDSF
                          COMMON/SAV3/GRO,GUESS,LP,NSOL,MSOL,NP,MOS,NMIS,NSPR,NPERPD(30),
ISN 0005                 1   PAD(30),LTR(50),PLR(50),RDIST(56,4),ALPI(4,60)
                          COMMON/SAVALL/LCK,SLO,NM,NEXD,NV,NUMD,MYRS,LZOPT(8),NYD(46),MAT(46
ISN 0006                 1),SUST(46),DS(46),LYD(46),YD(46),IS(102),LYR(252),LETT(250),
                         2  MIN(250),YRLM(250),VEH(4,60),NONREC(120,20),NMULT(60,50)
                          COMMON/TEMP/VNM(2,250),IFLAG,KI,NEXT,LOUT,SAVS(40),KOUT(40),
ISN 0007                 1  NINTYR(40,20),NTGYTR(40,20,2),RECUR(60,20,2)
                          COMMON/SCRACH/  DUMM(2057),RCOST(60),DUM(2291),VYTR(20,120)
ISN 0008            C     COUNT NUMBER OF EACH VEHICLE USED BY YEAR AND TEST RANGE
ISN 0009                  NV2 = 2*NV
ISN 0010                  DO 8032 I = 1,NV2
ISN 0011                  DO 8032 J = 1,MYRS
ISN 0012               8032 VYTR(J,I) = 0.0
ISN 0013                  DO 8033 L = 1,NM
ISN 0014                  IF(YRLM(L).LT..0001)  GO TO 8033
ISN 0016                  I1= MIN(L)
ISN 0017                  M = LETT(L)
ISN 0018                  I = I1
ISN 0019                  IF(LTR(M).EQ.2)  I = I1 & NV
ISN 0021                  X = NMULT(I1,M)
ISN 0022                  J = LYR(L)
ISN 0023                  VYTR(J,I) = VYTR(J,I) & YRLM(L)*X
ISN 0024               8033 CONTINUE
                    C     DETERMINE 'AVERAGE' RECURRING COST OF EACH VEHICLE
ISN 0025                  DO 8034 I = 1,NV
ISN 0026                  RCOST(I) = 0.0
ISN 0027                  TVEH = 0.0
ISN 0028                  I1 = I  & NV
ISN 0029                  DO 8035 J = 1,MYRS
ISN 0030                  RCOST(I) = RCOST(I) & VYTR(J,I)*RECUR(I,J,1) & VYTR(J,I1)*
                         1   RECUR(I,J,2)
ISN 0031                  TVEH = TVEH & VYTR(J,I) & VYTR(J,I1)
ISN 0032               8035 CONTINUE
ISN 0033                  IF(TVEH.LT..0001)  GO TO 8034
ISN 0035                  RCOST(I) = RCOST(I)/TVEH
```

```
ISN 0036        8034 CONTINUE
ISN 0037          99 RETURN
ISN 0038             END

****** END OF COMPILATION ******
```

```
F88-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED LIST,XREF,MAP,NCAL
          VARIABLE OPTIONS USED - SIZE=(126976,24576)          DEFAULT OPTION(S) USED
IEW0000       NAME MOX02VC(R)

                                   CROSS REFERENCE TABLE

CONTROL SECTION                    ENTRY
  NAME     ORIGIN  LENGTH          NAME  LOCATION    NAME  LOCATION    NAME  LOCATION    NAME
VEHRC        00      3D2
SAV3         3D8     980
SAVALL       D58     3A1C
TEMP         4778    4110
SCRACH       8888    6A60

 LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION      LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION
      E0          SAV3          SAV3                    E4          SAVALL          SAVALL
      E8          SAVALL        SAVALL                  EC          TEMP            TEMP
      F0          SCRACH        SCRACH                  F4          SCRACH          SCRACH
      F8          SCRACH        SCRACH
ENTRY ADDRESS     00
TOTAL LENGTH    F2E8

****MOX02VC    NOW REPLACED IN DATA SET
```